Computational models to investigate binding mechanisms of regulatory proteins

DISSERTATION

zur Erlangung des akademischen Grades

doctor rerum naturalium

(Dr. rer. nat.)

im Fach Informatik

eingereicht an der

Mathematisch-Naturwissenschaftlichen Fakultät

Humboldt-Universität zu Berlin

von

Dipl.-Inf. Alina-Cristina Munteanu (Gozman)

Präsidentin der Humboldt-Universität zu Berlin:

Prof. Dr.-Ing. Dr. Sabine Kunst

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät:

Prof. Dr. Elmar Kulke

Gutachter/innen:

1. Prof. Dr. Uwe Ohler
2. Prof. Dr. Ulf Leser
3. Prof. Dr. Raluca Gordân

Tag der mündlichen Prüfung: 11. Oktober 2017

Abstract

There are thousands of eukaryotic regulatory proteins that bind to specific cis regulatory regions of genes and/or RNA transcripts and coordinate gene expression. At the DNA level, transcription factors (TFs) modulate the initiation of transcription, while at the RNA level, RNA-binding proteins (RBPs) regulate every aspect of RNA metabolism and function. The DNA or RNA targets and/or the sequence preferences of hundreds of eukaryotic regulatory proteins have been determined thus far using high-throughput *in vivo* and *in vitro* experiments, such as chromatin immunoprecipitation (ChIP) followed by sequencing (ChIP-seq) and protein binding microarrays (PBMs) for TFs, or cross-linking and immunoprecipitation (CLIP) techniques and RNAcompete for RBPs. However, the derived short sequence motifs do not fully explain the highly specific binding of these regulatory proteins.

In order to improve our understanding of how different proteins achieve their regulatory specificity, we developed two computational tools that incorporate additional information in the analysis of experimentally determined binding sites. For protein-DNA interactions, we investigate the binding specificity of paralogous TFs (i.e. members of the same TF family). Focusing on distinguishing between genomic regions bound *in vivo* by pairs of closely-related TFs, we developed a classification framework that identifies putative co-factors that provide specificity to paralogous TFs. For protein-RNA interactions, we investigate the role of RNA secondary structure and its impact on binding-site recognition. We developed a motif finding algorithm that integrates secondary structure together with primary sequence in order to better identify binding preferences of RBPs.

Zusammenfassung

Es gibt tausende regulatorische Proteine in Eukaryoten, die spezifische cis-regulatorischen Elemente von Genen und/oder RNA-Transkripten binden und die Genexpession koordinieren. Auf DNA-Ebene modulieren Transkriptionsfaktoren (TFs) die Initiation der Transkription, während auf RNA-Ebene RNA-bindende Proteine (RBPs) viele Aspekte des RNA-Metabolismus und der RNA-Funktion regulieren. Für hunderte dieser regulatorischer Proteine wurden die gebundenen Gene beziehungsweise RNA-Transkripte, sowie deren etwaige Sequenzbindepräferenzen mittels *in vivo* oder *in vitro* Hochdurchsatz-Experimente bestimmt. Zu diesen Methoden zählen unter anderem Chromatin- Immunpräzipitation (ChIP) gefolgt von Sequenzierung (ChIP-seq) und Protein Binding Microarrays (PBMs) für TFs, sowie Cross-Linking und Immunpräzipitation (CLIP)-Techniken und RNAcompete für RBPs. In vielen Fällen kann die zum Teil hohe Bindespezifität für ein zumeist sehr kurzes Sequenzmotiv regulatorischer Proteine nicht allein durch die gebundene Primärsequenz erklärt werden.

Um besser zu verstehen, wie verschiedene Proteine ihre regulatorische Spezifität erreichen, haben wir zwei Computerprogramme entwickelt, die zusätzliche Informationen in die Analyse von experimentell bestimmten Bindestellen einbeziehen und somit differenziertere Bindevorhersagen ermöglichen. Für Protein-DNA-Interaktionen untersuchen wir die Bindungsspezifität paraloger TFs (d.h. Mitglieder der gleichen TF-Familie). Mit dem Fokus auf der Unterscheidung von genomischen Regionen, die *in vivo* von Paaren eng miteinander verwandter TFs gebunden sind, haben wir ein Klassifikationsframework entwickelt, das potenzielle Co-Faktoren identifiziert, die zur Spezifität paraloger TFs beitragen. Für Protein-RNA-Interaktionen untersuchen wir die Rolle von RNA-Sekundärstruktur und ihre Auswirkung auf die Auswahl von Bindestellen. Wir haben einen Motif-Finding-Algorithmus entwickelt, der Sekundärstruktur und Primärsequenz integriert, um Bindungspräferenzen der RBPs besser zu bestimmen.

Acknowledgements

The sinuous path of my research endeavors started in Romania at the "Al. I. Cuza" University of Iași, continued at Duke University, Durham, US, and finally led me to Max Delbrück Center, Berlin, Germany. Altough it was not a straightforward journey, I am gratefull for the opportunity to meet and work with a number of extraordinary people. The work and results reported here would not have been possible without their encouragement, guidance and support.

First of all, I would like to thank my advisors Prof. Liviu Ciortuz in Romania, Prof. Raluca Gordân in Durham and Prof. Uwe Ohler in Berlin. Prof. Ciortuz inspired and motivated me greatly to work in this field, and his assistance and advice were crucial to my research path. I am deeply indebted to Prof. Raluca Gordân for her patient guidance, enthusiastic encouragement and valuable knowledge and experience. Her support went well beyond my research project. I wish to show my greatest appreciation to Prof. Uwe Ohler, who gracefully hosted me for four years altought I initially came to his group for four months. His pertinent guidance and stimulating ideas have been invaluable. I would also like to thank Prof. Ulf Leser for accepting to review this thesis and host my enrolment at Humboldt-Universität zu Berlin, as well as for the critical assessment of my work.

I would like to acknowledge colleagues and friends from different places that contributed to inspiring and enjoyable working environments. I appreciate the help and inspiration provided by past and present members of the Ohler group, in particular Neelanjan Mukherjee, Hans-Hermann Wessels, Stoyan Georgiev, Dina Hafez and Rebecca Worsley-Hunt. Also, I miss the group from Iași that included Andrei Arusoaie, Lucian Gâdioi and Monica Boțoiu and our lunches and board-game evenings. I am grateful to a number of people that helped me proofread this thesis: Rebecca Worsley-Hunt, Neelanjan Mukherjee, Philipp Boss. I am also indebted to my german colleagues that were kind enough to translate the abstract: Hans-Hermann Wessels, Martin Burkert and Henriette Miko.

My husband Dan greatly supported my work by constant encouragements and confidence in me, and also by tolerating the sometimes quite demanding work times and all our vacations that were affected by dead-lines. He also invested some of his time and expertise in building the webserver for one of my projects. I can not thank him enough!

The DAAD (Deutscher Akademischer Austausch Dienst) grant A/12/84763 provided funding for part of the research presented in the following. Thank you very much!

Contents

Abstract iii

Zusammenfassung v

Acknowledgements vi

Contents vii

List of Figures xi

List of Tables xiii

Abbreviations & Symbols xv

1 Introduction 1
- 1.1 Contributions and goals . . . 3
- 1.2 Thesis outline . . . 5
- 1.3 Own prior work and contributions . . . 5

2 Background in molecular biology 7
- 2.1 From DNA to proteins . . . 7
  - 2.1.1 DNA . . . 7
  - 2.1.2 RNA . . . 9
  - 2.1.3 Nucleic acids structure . . . 9
  - 2.1.4 Gene expression . . . 11
  - 2.1.5 Proteins and gene regulation . . . 14
- 2.2 Detecting protein-DNA interactions . . . 16
  - 2.2.1 PBM experiments . . . 16
  - 2.2.2 ChIP-seq experiments . . . 17
- 2.3 Detecting protein-RNA interactions . . . 18
  - 2.3.1 SELEX experiments . . . 18
  - 2.3.2 RNAcompete experiments . . . 19
  - 2.3.3 CLIP-based experiments . . . 19

3 Background in bioinformatics 21
- 3.1 Processing experimental data . . . 21
  - 3.1.1 Processing microarray assays . . . 21

3.1.2 Processing genome-wide sequencing assays . . . 23
3.2 Predicting RNA structure . . . 24
3.2.1 Tools based on free energy minimization . . . 26
3.2.2 Tools based on ensembles of secondary structures . . . 26
3.3 Identifying the target sites of RBPs . . . 27
3.3.1 Tools based on sequence . . . 27
3.3.2 Tools based on sequence and structure . . . 29
3.4 Machine learning techniques . . . 32
3.4.1 Classifiers . . . 34
3.4.2 Performance evaluation . . . 44

4 Data sources **47**
4.1 Protein-DNA data . . . 47
4.1.1 In vitro datasets . . . 47
4.1.2 In vivo datasets . . . 48
4.2 Protein-RNA data . . . 49
4.2.1 In vitro datasets . . . 49
4.2.2 In vivo datasets . . . 49
4.3 Other data . . . 50
4.3.1 RNA-seq datasets . . . 50

5 Case study: differential *in vivo* DNA binding of paralogous TFs **53**
5.1 Introduction . . . 53
5.1.1 Myc/Max/Mad family . . . 53
5.2 Analysis . . . 54
5.2.1 Experimental data . . . 54
5.2.2 Overlap analysis . . . 57
5.2.3 Analysis using intrinsic DNA binding preferences . . . 58
5.3 Discussion . . . 61

6 COUGER: a tool to investigate protein-DNA interactions **63**
6.1 Introduction . . . 63
6.2 The COUGER framework . . . 63
6.2.1 Classification algorithms . . . 64
6.2.2 Classes . . . 64
6.2.3 Features . . . 65
6.2.4 Feature selection . . . 67
6.2.5 Parameter optimization . . . 72
6.2.6 Performance estimation . . . 74
6.2.7 Server design . . . 75
6.2.8 Additional analyses . . . 75
6.3 Results . . . 78
6.3.1 ENCODE ChIP-seq datasets . . . 78
6.3.2 Classification performance . . . 79
6.3.3 Identification of putative co-factors . . . 80
6.3.4 Classification between replicate experiments . . . 82
6.4 Discussion . . . 83

7 SSMART: an algorithm for *de novo* RNA motif identification 87
7.1 Introduction . . . 87
7.2 Methods . . . 89
7.2.1 RNA secondary structure prediction . . . 89
7.2.2 The sequence-structure motif identification framework . . . 93
7.2.3 Synthetic datasets . . . 99
7.2.4 Experimental datasets . . . 100
7.2.5 Evaluation on synthetic datasets . . . 102
7.2.6 Evaluation on CLIP datasets . . . 103
7.2.7 Parameter optimization . . . 104
7.3 Results . . . 104
7.3.1 Recovering sequence and structure motifs from synthetic datasets . 105
7.3.2 Testing motif predictions on CLIP datasets . . . 107
7.3.3 Identification of motifs from *in vivo* and *in vitro* datasets . . . 107
7.4 Discussion . . . 111

8 Conclusions and outlook 113
8.1 Distinguishing between TF-DNA interactions for TFs with similar binding motifs . . . 113
8.2 Identifying the binding motif for RBP-RNA interactions . . . 114

Bibliography 117

List of Figures

1.1 Overview 4

2.1 Nucleic acids & protein structures 10
2.2 RNA secondary structures 11

3.1 Visualizing microarray data 22
3.2 Visualizing high-throughput sequencing data 25
3.3 Support vector machines 34
3.4 Support vector machines and kernels 38
3.5 Decision trees 40

5.1 DNA binding motifs for Myc, Max, and Mad in Transfac 54
5.2 DNA binding motifs for Myc, Max, and Mad from PBM data 57
5.3 Overlap between c-Myc and Mad ChIP-seq peaks 58
5.4 AUC enrichments of motifs in c-Myc and Mad datasets 60
5.5 Myc and Mad fractions of ChIP-seq and DNase-seq peaks 62

6.1 The **COUGER** framework 64
6.2 Illustration of **COUGER**'s feature derivation 66
6.3 Peak caller comparison for c-Myc dataset 77
6.4 **COUGER** classification performance with PBM features 79
6.5 **COUGER** classification performance with PBM and PWM features . . . 80
6.6 **COUGER** output 85

7.1 The **SSMART** framework 89
7.2 Structural predictions with RNAplfold 91
7.3 Structural prediction: RNAprofiling vs RNAplfold 93
7.4 **SSMART** visualization of motif clusters 99
7.5 Structural environments in synthetic data 101
7.6 Predictions on synthetic datasets 103
7.7 Comparison with other tools on synthetic datasets 106
7.8 Comparison with other tools on biological datasets 108
7.9 **SSMART** results on biological data: *in vivo* vs. *in vitro* 109
7.10 **SSMART** motifs for selected CLIP datasets 110

List of Tables

2.1 Standard genetic code 13

3.1 Motif finding algorithms used to analyse RBP-RNA interactions 28

4.1 TFs in UniPROBE by species 48
4.2 TFs in ENCODE by cell type 48
4.3 RBPs in CISBP-RNA by species 49
4.4 CLIP datasets 50
4.5 RNA-seq in ENCODE 51

5.1 Number of ChIP-seq peaks for Myc, Max and Mad 55
5.2 (Not)overlapping sequences for Myc and Mad 57

6.1 Features selected by various FS algorithms 69
6.2 Features selected by mRMR with simple discretizations 71
6.3 Complex discretization intervals 73
6.4 Features selected by mRMR with complex discretizations 74
6.5 Pairs of paralogous TF used in **COUGER** 78

7.1 Sequence motifs in synthetic data 101
7.2 Biological datasets 102
7.3 Cutoffs for synthetic datasets 103
7.4 CLIP datasets used to compare different motif finders 104
7.5 Motif recovery on synthetic datasets 105

Abbreviations & Symbols

Mathematical abbreviations

AUC	area under the ROC curve
CART	classification and regression trees
CBFS	clearness-based feature selection
CM	covariance model
CV	cross-validation
E-score	enrichment score
EM	expectation-maximization algorithm
FS	feature selection
HMM	hidden Markov model
IDR	irreproducible discovery rate
LOOCV	leave-one-out cross-validation
mRMR	minimum redundancy maximum relevance feature selection
MI	mutual information
NMIFS	normalized mutual information feature selection
OOB	out-of-bag / out-of-bootstrap
PWM	position weight matrix
RF	random forest
ROC	receiver operating characteristic
RBF	redundancy-based feature selection
RF-FS	RF recursive feature elimination
$\mathbf{RF_{gi}}$	RF with the Gini index importance
$\mathbf{RF_{pi}}$	RF with the unscaled permutation importance
SCFG	stochastic context-free grammar

SVM	support vector machine
$\mathbf{SVM_{lin}}$	SVM with linear kernel
$\mathbf{SVM_{rbf}}$	SVM with radial basis function kernel
SVR	support vector regression

Biological abbreviations

4-SU	4-thiouridine
6-SG	6-thioguanosine
ChIP	chromatin immunoprecipitation
ChiP-seq	ChIP followed by high-throughput sequencing
CLIP	cross-linking and immunoprecipitation
CLIP-seq	high-throughput sequencing CLIP
DNA	deoxyribonucleic acid
DNase-seq	DNaseI digestion followed by high-throughput sequencing
dsRBD	dsRNA binding domain
dsRNA	double-stranded RNA
HITS-CLIP	see CLIP-seq
HLH	helix-loop-helix domain
HT-SELEX	hight-throughput SELEX
HTH	helix-turn-helix domain
iCLIP	individual-nucleotide resolution ultraviolet CLIP
KH	hnRNP K-homology domain
mRNA	messenger RNA
ncRNA	non-coding RNA
ORF	open reading frame
PAR-CLIP	photo-activatable ribonucleoside enhanced CLIP
PBM	protein binding microarray
PCR	polymerase chain reaction
pre-mRNA	primary transcript mRNA
RBP	RNA-binding protein
RNA	ribonucleic acid
RNA-seq	RNA sequencing

RRM	RNA recognition motif
rRNA	ribosomal RNA
SELEX	systematic evolution of ligands by exponential enrichment
ssRNA	single-stranded RNA
TF	transcription factor
tRNA	transfer RNA
TSS	transcription start site
uPBM	universal PBM
UTR	untranslated region
UV	ultra violet
Zip	leucine zipper domain
ZnF	zinc finger domain

IUPAC codes

Symbol	Bases represented	Description
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine
U	U	Uracil
W	A or T	Weak
S	C or G	Strong
M	A or C	aMino
K	T or G	Keto
R	A or G	puRine
Y	C or T	pYrimidine
B	C or T or G	not A (B comes after A)
D	A or T or G	not C (D comes after C)
H	A or T or C	not G (H comes after G)
V	A or C or G	not T (V comes after T and U)
N	A or C or G or T	aNy base (not a gap)

To my son Victor
you bring me joy and make me proud

Chapter 1

Introduction

Gene regulation is a multiple levels process that fine tunes the expression of genes so that the concentration of macromolecular components meets the exact needs of the cell(s). Although the same complete copy of deoxyribonucleic acid (DNA) is present in each cell of an organism, the amount and type of ribonucleic acid (RNA) and proteins varies, thus enabling the cells to have different appearance and properties (phenotype) and to respond to changes in environmental conditions. Transcription factors (TFs) work at the DNA level by activating or repressing the initiation of transcription, while RNA-binding proteins (RBPs) work at the RNA level and regulate every aspect of RNA metabolism and function. Both types of regulatory proteins bind to specific sites in cis regulatory regions of genes/transcripts. At the cellular level, these molecular interactions coordinate not only the expression of genes, but also the phenotypic fate of the cell.

In order to better understand the activity and specificity of proteins that bind nucleic acids (TFs and RBPs), we need to determine and characterize the sites they recognize. The protein-DNA or protein-RNA interactions for hundreds of eukaryotic proteins were studied so far with high-throughput *in vitro* and *in vivo* techniques, like protein binding microarrays (PBMs) [11] and chromatin immunoprecipitation (ChIP) coupled with sequencing (ChIP-seq) [62] for protein-DNA interactions and RNAcompete [120] and a number of crosslinking and immunoprecipitation (CLIP) methods [53, 76, 145] for protein-RNA interactions. Although a lot in known about these interactions, our understanding is incomplete for several reasons. First, the experimental protocols are prone to various inherent biases and the measurements have a certain amount of noise. For example, the ChIP-seq method was shown to report consistent binding of unrelated TFs to highly expressed genes in yeast [113, 141]. Second, the computational approaches are based on imperfect models for binding specificity. For instance the position weight matrix (PWM) models the probability to have each of the four possible nucleotides

at each position in the binding motif, but assumes that each position is independent. Despite this, the advantages of this representation are that it is easy to use, visualize and understand. Third, it is challenging to account for the complexity of biological interactions. Most of the current knowledge is based on individual analyzes of different types of experimental data. The focus now is to integrate several categories of available information.

Even if the genetic code was deciphered some time ago and scientists from various fields are working since then to understand the regulatory code, some details of gene regulations are still not elucidated. For example, not much is known about why only some of the genome-wide putative and accessible binding locations are actually bound *in vivo* by a particular protein. Even if the TF-DNA interactions are well studied, one challenge is to distinguish between the genome-wide binding of proteins with similar binding preferences. Many TFs share not only the same binding domain, but also the DNA sequences they recognize and bind to [6]. Therefore, I investigated the binding specificity of paralogous TFs (i.e. members of the same TF family) and I focused on distinguishing between genomic regions bound *in vivo* by pairs of closely-related TFs. To this end I developed **COUGER** (co-factors associated with uniquely-bound genomic regions), a classification framework that identifies putative co-factors that provide specificity to paralogous TFs. I am not aware of other methods that perform similar or identical computations. Related tools that identify differentially enriched DNA motifs are available (e.g., the MEME software suite [8]). However, such tools search for one DNA motif at a time. In contrast, **COUGER** analyzes hundreds of motifs from potential co-factors and selects a small set of TF motifs that are used to correctly classify between target regions of pairs of TFs.

In the case of protein-RNA interactions, the binding preferences consist not only in the sequence an RBP recognizes, but also in the associated RNA structure [20]. Although many known RBPs prefer to bind ssRNA regions and some *de novo* motif finders designed for TF-DNA interaction were successfully applied on RNA binding data [8, 40, 46], there is a need for identifying the sequence-structure motif of RBPs. In recent years were developed multiple tools that derive RBP binding motifs accounting for the RNA structure [7, 66, 97], but some of them are designed for data from a specific type of experiment, while others work in a classification setting. I propose **SSMART** (sequence-structure motif analysis tool for RNA-binding proteins), a computationally efficient *de novo* motif discovery tool based on analyzing RNA quantitative regulatory evidence from high-throughput experimental data. It models the RBP binding preferences for RNA's sequence and structure, extending the conserved Evidence Ranked Motif Identification Tool (cERMIT) [46]. By optimizing a rank-based enrichment function, **SSMART** easily handles transcriptome-wide scale data, which is not the case for many motif finders.

SSMART performs favorably to existing methods in thorough benchmarking of existing tools using synthetic data with varying amounts of sequence and structure content. It was also successfully used on high-throughput *in vitro* and *in vivo* RBP-RNA interaction datasets, recovering the known sequence motifs.

1.1 Contributions and goals

The central theme of this thesis is the study of interactions between regulatory protein and nucleic acids combining high-throughput experimental data with an additional layer of information. In particular, I focus on two main problems:

1. distinguishing between TF-DNA interactions for TFs with similar binding motifs and
2. identifying the sequence-structure binding motifs for RBP-RNA interactions.

As the first main contribution I present **COUGER**, the first computational tool (to my knowledge) for studying differential genomic binding by paralogous TFs. TFs from the same protein family often have highly similar DNA binding preferences, but they perform different functions and bind to different sets of genomic regions *in vivo*, as observed from ChIP-seq experiments. **COUGER** uses a classification approach to identify a small set of DNA motifs, belonging to putative co-factors, that best explain the differences between two given sets of DNA sequences (Fig. 1.1A). Specific contributions to the objective of distinguishing between binding of paralogous TFs are summarized below:

- A framework is established for classifying specific sets of genomic regions bound *in vivo* by pairs of TFs.
- The binding motifs of potential co-factors are taken into account. The proposed approach uses available PWM or PBM data for hundreds of TFs.
- A custom feature selection procedure reduces the number of putative co-factors to less than 10 without a significant decrease in classification accuracy.
- A highly interactive web server allows users to display details of the overall classification performance, the performance of individual runs in the cross-validation process, as well as the results of the various feature selection procedures.
- The generality of the model allows its application to other biological questions. The two given classes of DNA sequences may represent other biological properties.

As the second main contribution I present **SSMART**, a *de novo* motif discovery tool for sequence-structure motifs of RBPs. Even if recent methodological advances have enabled high-throughput determination of both *in vitro* and *in vivo* binding specificity of RNA binding proteins, for most RBPs only a primary sequence motif has been determined,

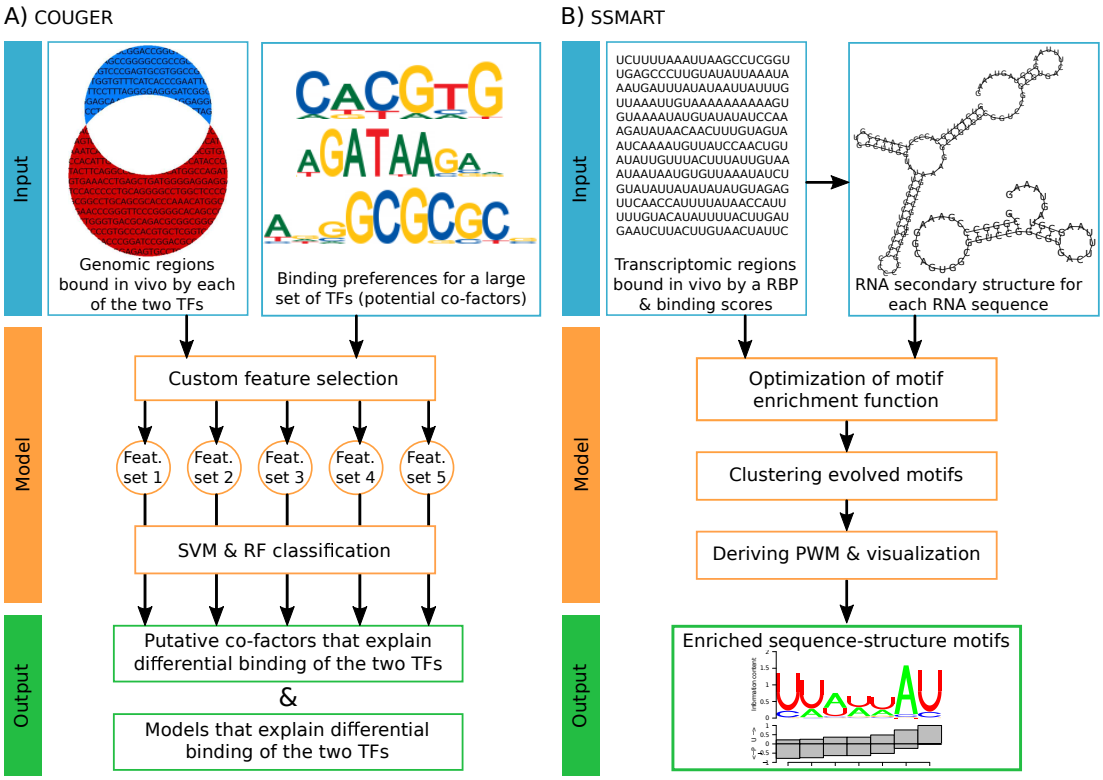


FIGURE 1.1: Overview of the two proposed methods. A) The **COUGER** framework for distinguishing between TF-DNA interactions for TFs with similar binding motifs. B) The **SSMART** framework for *de novo* identification of the sequence-structure binding motifs for RBP-RNA interactions.

while the structure of the binding sites remains largely uncharacterized. **SSMART** identifies binding preferences from different types of experimental data by searching for optimal sequence-structure motifs of flexible lengths (Fig. 1.1B). Specific contributions to RNA motif finding are summarized below:

- The RNA secondary structure is taken into account. The structural predictions are made with RNAprofiling, a tool based on an ensemble of secondary structures [128], but **SSMART** is not restricted to this method of obtaining the folding of RNA sequences.
- A representation is proposed that simultaneously models the primary sequence and the secondary structure of the RNA binding site. The sequence-structure model uses an extended alphabet and is easy to use, visualize and interpret.
- The generality of the model allows its application to different types of experimental data, provided that an appropriate score for the binding strength can be computed.

1.2 Thesis outline

This work is structured as follows:

Chapter 2 provides the necessary molecular biology background. I describe the flow of information in the cell and then I summarize the experimental methods for studying interactions between proteins and nucleic acids.

Chapter 3 describes bioinformatics techniques related to this work. I present the methods used to process the experimental data related to protein-nucleic acids interactions. Then I describe tools used for RNA secondary structure prediction and for identification of RNA-binding proteins motifs. Finally I present a selection of basic machine learning techniques.

Chapter 4 gives an overview of the available databases for protein-DNA and protein-RNA interactions.

Chapter 5 presents a case study involving the differential *in vivo* DNA binding of paralogous transcription factors. Using experimental data for Myc/Max/Mad family of TFs, I investigate whether the intrinsic DNA binding preferences can explain why paralogous TFs interact with different genomic sites *in vivo*.

Chapter 6 addresses the problem of distinguish between genomic regions bound uniquely by closely related TFs. I describe **COUGER**, a novel classification framework that identifies sets of putative co-factors for pairs of paralogous TFs.

Chapter 7 turns to the problem of motif finding in the context of RBPs. I present **SSMART**, a tool that models secondary structure information together with primary sequence and identifies sequence-structure motifs for RBPs.

Chapter 8 discusses the main outcomes of this thesis and some aspects concerning possible future work.

1.3 Own prior work and contributions

Chapter 5 of this thesis presents the case study described in [105]. The authors' roles can be assigned as follows: Raluca Gordân (Duke University, Durham, USA) designed the workflow and Alina Munteanu performed all associated analysis.

Chapter 6 describes the classification framework **COUGER** initially proposed in [105] and further extended in [107]. The contributions can be attributed to the authors as follows: Alina Munteanu and Raluca Gordân designed the classification framework.

Alina Munteanu implemented the tool and performed all associated analyzes. Alina Munteanu and Dan Munteanu (Max Delbrück Center, Berlin, Germany) developed the web server. Raluca Gordân interpreted the results for different DNA-binding proteins and together with Uwe Ohler (Max Delbrück Center, Berlin, Germany) advised on the project.

Chapter 7 presents the motif finder **SSMART** described in [106]. The roles of the authors, Alina Munteanu, Neelanjan Mukherjee and Uwe Ohler, can be assigned as follows: Alina Munteanu and Uwe Ohler designed the motif finding strategy for sequence-structure motifs (**SSMART**). Alina Munteanu implemented the tool, using the existing algorithm for sequence motifs (cERMIT), and performed all associated analyzes. Neelanjan Mukherjee (Max Delbrück Center, Berlin, Germany) interpreted the results for different DNA-binding proteins and together with Uwe Ohler advised on the project.

Chapter 2

Background in molecular biology

This chapter describes some concepts and methods from molecular biology that provide the necessary background for the work presented in this thesis. I start with a general description of gene expression and gene regulation, then follow with introducing some experimental techniques that detect interactions between proteins and nucleic acids.

2.1 From DNA to proteins

Hereditary information of an organism is stored and transmitted whithin cells by three classes of biomolecules: deoxyribonucleic acid (DNA), ribonucleic acid (RNA) and protein [1]. These macromolecules are polymers: long and unbranched chains of subunits, that form complex tree-dimensional structures. The dynamic flow of information between them is summarized by the central dogma of molecular biology [23]:

$$\text{DNA} \xrightarrow{Transcription} \text{RNA} \xrightarrow{Translation} \text{protein}$$

The following sections describe these macromolecules and their interplay in the context of eukaryotic cells [81], with a focus on the concepts that are used in later chapters.

2.1.1 DNA

DNA (deoxyribonucleic acid) is a nucleic acid macromolecule that stores the genetic information in the cell. The basic units of DNA are called deoxyribonucleotides and are formed from a sugar (deoxyribose) bound on one side to a phosphate group and on the other side to one of the four possible nitrogenous bases: adenine (A), cytosine (C),

guanine (G) and thymine (T) [5]. Adenine and guanine are purines and have double-ring structures with a six-membered and a five-membered ring containing nitrogen, while cytosine and thymine are pyrimidines and have a single six-membered ring containing nitrogen. The nucleotides are linked together by covalent bonds between the sugar-phosphates and form a long polymeric chain (see Figure 2.1A: primary level). A DNA molecule is composed from two such chains that are bonded together and form a helical three-dimensional structure. The sugar-phosphate backbones are on the outside of the double helix, while the bases are on the inside and form hydrogen bonds between specific pairs. The hydrogen bonds occur between a large purine base (A or G) and a small pyrimidine base (T or C). There are two hydrogen bonds between A and T and three hydrogen bonds between G and C. These two pairs are called Watson-Crick base pairs and their respective bases are complementary bases [147]. The complementarity of the two strands of the DNA molecule means that both strands carry the same genetic information. This gives not only increased stability to the molecule, but provides also a replication mechanism. In order to copy the hereditary information (for example in cell division), the double helix separates and each individual strand is used as a template in the synthesis of a new complementary strand [1].

The phosphate end of a DNA strand is called the 5' end, while the deoxyribose end is called the 3' end. The DNA molecule is read by the transcriptional machinery in a particular direction, and the strand that is traversed from its 5' end to its 3' end is called the sense strand, while the complementary strand is called the anti-sense strand. The genetic information of a DNA molecule is represented as a sequence of letters from the 4-letter alphabet {A, C, G, T}, written from left to right, that correspond to the nucleotides of the sense strand. Sometimes an extended alphabet is used to denote not only exact bases, but also possible subsets from the set of all four, for example R for a puRine (A or G) or N for aNy base (see IUPAC codes). The distance in a DNA sequence is measured in bases or base pairs (bp). A nucleotide situated on the left (the 5' side) of a reference location is said to be upstream, and one located on the right (the 3' side) is called downstream [1].

All DNA molecules of a cell form the genome. The prokaryotic cells do not have a nucleus and their genome consists in a single DNA molecule that has the double stranded helix structure. They are also haploid, i. e. have only one copy of their genetic information. In eukaryotes, the DNA is divided into several molecules, called chromosomes, and these chromosomes are usually found in multiple copies in a cell [102]. Vertebrates are polyploid, having at least one set of chromosomes inherited from each parent.

The chromosomes are packed in higher-level structures in order to fit in the cell. In humans, for example, a linear DNA helix that contains the whole genome would be

two meters long. The DNA double helix is first compacted with the help of proteins called histones, which form cylindrical complexes. DNA is coiled up around the histones, forming nucleosomes (see Figure 2.1A: quaternary level). This DNA-histone complex is called chromatin (an 11 nm fiber) and has the appearance of "beads-on-a-string". DNA is folded and compacted further at the level of linker DNA between nucleosomes, and at the level of chromatids to form a 30 nm fiber [1]. The tight packing of DNA does not serve only the space fitting purpose, but has other important functions, like preventing DNA damage. DNA folding is a dynamic process that depends on the cell cycle and other cell necessities. For example, the local structure regulates the accessibility of specific regions in the DNA molecule to the transcriptional machinery [148].

2.1.2 RNA

RNA (ribonucleic acid) is a nucleic acid macromolecule. Like DNA, RNA is also a biopolymer with a very similar basic unit – a sugar with a phosphate group on one side and a nitrogenous base on the other – called ribonucleotide. There are two main differences between DNA and RNA nucleotides: the sugar has an extra hydroxyl group and is called ribose (hence the name *ribo*nucleic acid), and thymine is replaced by uracil (U), a related base that pairs also with adenine. The modified sugar makes RNA less stable than DNA [1].

Even if the chemical differences between DNA and RNA are small, their overall structure is quite different [125]. While DNA molecules contain millions of nucleotides (a human chromosome can have up to 250 million bp), RNA molecules are much shorter, containing just hundreds or thousands of nucleotides, and many are much shorter. Also, whereas DNA always occurs as a double-stranded helix, an RNA molecule is usually formed from a single strand of nucleotides. A single RNA strand does not remain in a linear form, but folds on itself by local complementary base-pairing hydrogen bonds. The interactions that occur in RNA are the Watson-Cricks pairs: C-G and A-U, but also some other non-canonical pairings, called wobble base pairs, like G-U, or I-U, I-A, I-C, where I denotes inosine (a modified adenine that is read as a guanine).

2.1.3 Nucleic acids structure

While the linear sequence of nucleotides that are linked together by phosphodiester bonds is called the primary structure, the set of base pairs that can be mapped on a single plane form the secondary structure of both the RNA and DNA molecule (Figure 2.1A: secondary level). In the case of DNA, the secondary structure is the "ladder" created by the hydrogen bonds between the two strands of nucleotides. For the single strand

of RNA nucleotides, there are various possibilities to form hydrogen bonds between bases and fold, resulting in a mix of single-stranded and double-stranded regions of the molecule [38]. The basic elements in RNA secondary structure are helices, loops and bulges (Figure 2.2). The most common structural element is the stem-loop, or the hairpin-loop, which appears when the RNA strand folds back on itself. The double-stranded region is called the stem, while the single-stranded nucleotides at one end of the stem form the loop or the hairpin. If a helix is separated on one strand by unpaired nucleotides it forms a bulge, and if the separation is on both strands it forms an internal loop. There are also more complex secondary structures called pseudoknots, that contain at least two stem-loop structures in which part of one stem is intercalated between the two parts of another stem. Computationally, their base pairing is not well nested and they are difficult to predict.

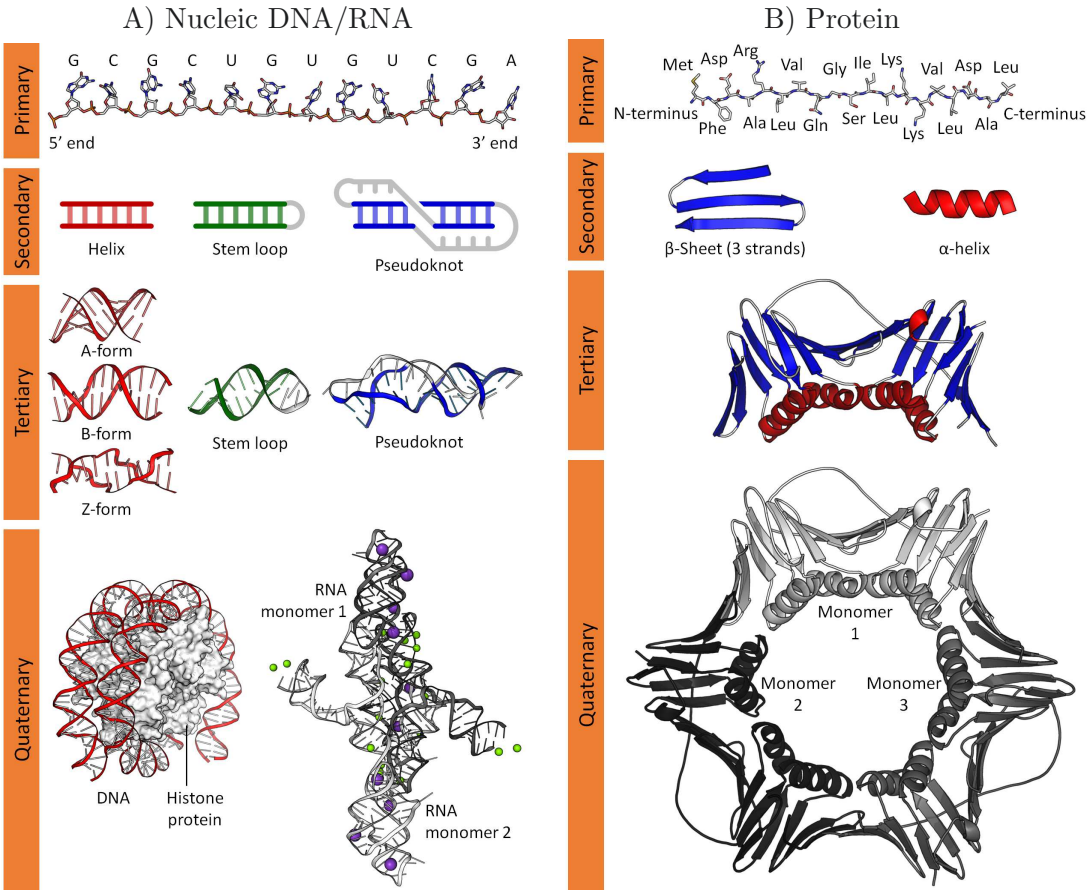


FIGURE 2.1: Different levels of structure in nucleic acids (A) and protein (B) molecules: primary, secondary, tertiary, and quaternary. Panel A contains DNA helices and examples from the VS ribozyme and telomerase and nucleosome, and Panel B shows PCNA - Proliferating cell nuclear antigen. Figures by Thomas Shafee [CC-BY-4.0], via Wikimedia Commons.

The tertiary structure of nucleic acids molecules corresponds to the three-dimensional arrangement of their atoms. The dominant tertiary structure is the double helix, which

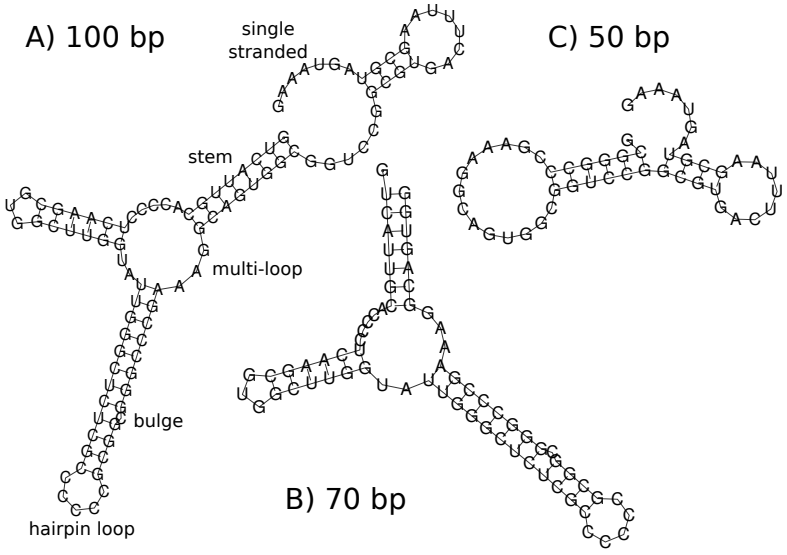


FIGURE 2.2: Examples of RNA secondary structure in the form of secondary structure diagrams. The folding and plots were done with RNAfold from ViennaRNA package [90]. The corresponding RNA sequences are all substrings of the same sequence, but with different lenghts.

is present in the DNA in three naturally occurring conformations: A-DNA, B-DNA, and Z-DNA (Figure 2.1A: tertiary level). The B-shape helix is dominant for DNA in cells and was first described by James D. Watson and Francis Crick [147]. It has a major groove and a minor groove and makes one complete turn about its axis every 10 bp of sequence. The double-helical RNA stems usually adopt an A-shape structure that has a deeper and narrower major groove.

A higher level of structural organization is the quaternary structure, which in the case of nucleic acids refers to the interactions between DNA molecules and proteins such as in chromatin, or between separate RNA molecules (Figure 2.1A: quaternary level).

2.1.4 Gene expression

The fundamental unit of heredity is called gene and represents a discrete segment of DNA that codes information for one specific (or more than one) RNA or protein molecule [114]. A protein is a folded polypeptide chain, i.e. a chain formed from amino acids. We focus next on the case of protein-coding genes (genes that encode for proteins) and their expression. The first step in expressing a gene is a process called transcription, in which the gene sequence is copied into an RNA molecule [1]. Transcription is performed in the nucleus by an RNA polymerase molecule. This enzyme recognizes the regulatory region of the gene, a promoter sequence in the DNA double helix, and binds to it. Then RNA polymerase breaks the hydrogen bonds between complementary deoxyribonucleotides and separates the two strands of the DNA helix. After the template strand is accessible,

the polymerase starts synthesizing RNA from the transcription start site (TSS). It moves along the DNA template strand from 3' to 5' appending ribonucleotides, that match the deoxyribonucleotides, into a growing RNA chain. The RNA is elongated one nucleotide at a time and the double stranded DNA helix reforms as the RNA molecule is released. The rapid release of the RNA strand from the DNA template allows for the synthesis of multiple RNA copies of the same gene in a short time.

The RNA molecule created by transcription is called the primary RNA transcript. The primary RNA transcript is converted into mature RNA by processes that take place in the cell nucleus during or immediately after transcription. For some genes, the mature RNA is the final product, but for the protein-coding genes, the corresponding RNA is called messenger RNA (mRNA) and it is used for the synthesis of protein molecules. There are four main modifications in the case of pre-mRNA: 1) capping, 2) splicing, 3) 3' end cleavage and 4) polyadenylation. The first modification is 5' capping, a process that starts during transcription, after around 25 nucleotides of mRNA are synthesized. In capping, a modified guanine nucleotide is added to the 5' end of the new mRNA molecule. This cap protects the mRNA from degradation and later plays a role in translation. It is also helping the cell distinguish between mRNAs and non-coding RNAs, that are not capped. The 3' end of the transcript is also processed. After translation, if the polyadenylation signal sequence is present in the nascent mRNA molecule, the mRNA is cleaved at a specific distance downstream of the signal and around 50-200 adenine nucleotides, depending on the organism, are enzymatically added by poly-A polymerase, without a DNA template. The poly(A) tail has similar functions to the 5' cap, protecting the mRNA from degradation and assisting in translation, but it also plays a role in export from the nucleus.

A major co-/post-transcriptional modification of the pre-mRNA molecule is splicing, a process in which some regions of pre-mRNA are removed. Most protein-coding genes contain both exons (protein-coding regions) and introns (non-coding regions) in an alternating pattern. Therefore the primary transcript mRNA has also intronic and exonic regions. This transcript is edited by splicing so that introns are eliminated and the exons are ligated together. A notable aspect of splicing is that it can produce different mature mRNA from the same transcript. This is a widespread phenomenon called alternative splicing, and the resulting transcripts are called alternatively spliced isoforms. RNA isoforms can be obtained in multiple ways: retaining introns, skipping or extending exons. The expression levels of different isoforms are not uniform and may also vary across tissues or cellular conditions [112].

The mRNAs are information carriers on the pathway to proteins, so the mature mRNA molecules are exported to the cytoplasm where translation takes place. Every mRNA

TABLE 2.1: Standard genetic code. The table contains all 64 RNA codons and their corresponding amino acids, represented by their standard 3 letter and 1 letter abbreviations. Three codons act as termination sites (stop codons), while one codon, AUG, acts as a initiation codon and also codes for methionine.

1st base	2nd base								3rd base
	U		C		A		G		
U	UUU	Phe/F	UCU	Ser/S	UAU	Tyr/Y	UGU	Cys/C	U
	UUC		UCC		UAC		UGC		C
	UUA	Leu/L	UCA		UAA	Stop	UGA	Stop	A
	UUG		UCG		UAG	Stop	UGG	Trp/W	G
C	CUU		CCU	Pro/P	CAU	His/H	CGU	Arg/R	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Gln/Q	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Ile/I	ACU	Thr/T	AAU	Asn/N	AGU	Ser/S	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lys/K	AGA	Arg/R	A
	AUG	Met/M	ACG		AAG		AGG		G
G	GUU	Val/V	GCU	Ala/A	GAU	Asp/D	GGU	Gly/G	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glu/E	GGA		A
	GUG		GCG		GAG		GGG		G

contains three parts: the 5' untranslated region (5'UTR), the protein-coding region or open reading frame (ORF), and the 3' untranslated region (3'UTR). As the name suggests, only the protein-coding region is translated. While transcription employs a direct one-to-one correspondence between DNA and RNA nucleotides, translation uses a more complex conversion, since there are only four different nucleotides in mRNA and twenty possible amino acids in proteins. The mRNA message is decoded in consecutive groups of three nucleotides, called codons [109]. Each codon specifies either one amino acid or a stop signal for the translation process, in a redundant manner. Since there are 64 possible combinations of three nucleotides, multiple codons represent the same amino acid. The alternative codons for an amino acid usually share the first two bases and differ only in the third nucleotide. The rules of translation are known as the genetic code (Table 2.1) [59].

The translation of an mRNA molecule into protein is performed by a large assembly called the ribosome. A ribosome is formed from proteins and RNA called ribosomal RNA (rRNA), and it is assisted in translation by a set of small RNA molecules that carry out the codon-amino acid correspondences from the genetic code. These molecules, called transfer RNAs (tRNAs), have only about 80 nucleotides in length and transport amino acids to the ribosome. All tRNAs fold in a specific secondary structure with four

short double helices and a central multiloop. One of the three hairpin loops contains the anticodon, a set of three consecutive nucleotides complementary to a RNA codon, while the single stranded 3' end binds to the specific amino acid. Translation starts with initiation, a short phase in which the ribosome assembles at the start codon of the target mRNA and the first tRNA is attached. Then comes the main phase, elongation: the codons are recognized one by one by the proper tRNA and a peptide bond is formed with the new amino acid. So the polypeptide grows as the ribosome translocate to the next codon. The final phase, termination, occurs when one of the three stop codons is reached. In this case, the ribosome releases the completed polypeptide. This process can be repeated multiple times before the mRNA molecule is degraded [1].

After translation, the polypeptide is folded into its specific three-dimensional structure. Similar to nucleic acids, proteins have four distinct structure levels (Figure 2.1B). The primary structure of a protein denotes the linear sequence of amino acids in the polypeptide chain, while the secondary structure refers to local sub-structures. There are two main types of secondary structure, the α-helix and the β-strand, that are both highly regular and saturate the hydrogen bond donors and acceptors in the peptide backbone. The tertiary structure depicts the three-dimensional compact globular structure of a protein molecule, while the quaternary structure refers to multi-subunit proteins. Complexes of two or more proteins called multimers are very common, and can contain identical subunits (e.g. the homotrimer from Figure 2.1B), or different subunits, such as hemoglobin – a heterotetramer with two α and two β subunits.

2.1.5 Proteins and gene regulation

Although the same complete copy of DNA (the genome) is present in each cell of an organism, the amount and type of RNA and proteins varies, thus enabling the cells to have different appearance and properties (phenotype) and to respond to changes in environmental conditions. In order to achieve this qualitative and quantitative variation in RNAs and proteins, genes are expressed only if/when the RNAs/proteins are needed. This fine-tuning is accomplished by gene regulation, a complex process that can involve all steps of gene expression: 1) transcription; 2) RNA processing; 3) RNA transport and localization; 4) translation; 5) RNA/protein degradation [81].

The first layer of gene regulation happens at the DNA level and controls transcription initiation. DNA packaging and DNA methylation play a role in transcriptional regulation, but the main players are gene regulatory proteins called transcription factors (TF). These proteins recognize and bind to specific regions of DNA and can stimulate transcription (transcriptional activators) or inhibit it (repressors). Some regulatory proteins

compete with each other for binding to specific regulatory sequences, while others need to dimerize or to cooperatively interact with specific protein partners in order to bind their DNA targets. The binding of proteins to DNA is based on the complementarity of the protein surface to the major or minor groove of the DNA in that region. The short stretches of DNA that are recognized by individual sequence specific DNA binding proteins or protein complexes have a common sequence composition and are called binding motifs. On the protein side, the interaction with DNA occurs through one of several possible domains incorporated in its tertiary structure. The common DNA binding domains include the helix-turn-helix (HTH), the homeodomain, the helix-loop-helix (HLH), the leucine zipper (Zip) and several types of zinc finger domains (ZnF). Each of these domains consists in a specific tertiary structure that have the same type of contact with the double helical DNA. For example the helix-turn-helix domain is constructed from two α helices connected by a short chain of aminoacids (the "turn") and held at a fixed angle. One of its helices, called the recognition helix, fits into the major groove of the DNA, while the other helps its positioning. The aminoacid composition of the recognition helix differs from protein to protein and gives the protein its binding specificity.

Another layer of gene regulation is carried out at the RNA level by RNA-binding proteins (RBPs), RNA helicases, RNA nucleases and non-coding RNA molecules, influencing the RNA metabolism and function (RNA processing, RNA transport and localization, RNA degradation). The RNA-binding proteins are cytoplasmic or nuclear proteins that bind to the double or single stranded RNA in cells [87]. Like DNA-binding proteins, many proteins that bind RNA have modular structures and are composed of a few modules of conserved structure [93]. The most common RNA-binding domains are RNA recognition motif (RRM), dsRNA binding domain (dsRBD), zinc finger (ZnF), and hnRNP K-homology domain (KH). For example, the RRM has a four-stranded β-sheet packed against two α-helices, while the classical zinc finger domain, found also in TFs, consists in a β-hairpin and an α-helix that are pinned together by a Zn^{2+} ion.

The interactions between proteins and RNA differ from those between proteins and DNA because of the different structure of the nucleic acids. RNA-binding proteins achieve their binding specificity through a mixture of sequence and structure properties, in variable proportions. For example, the majority of sequence-specific RBPs interact only with single-stranded RNA (ssRNA), and recognize the sequence of nucleotides. We note that the two most common RBDs in eukaryotes, the RNA recognition motif and the KH domains bind single-stranded RNA. But even if their binding site is single-stranded, the structural context can be different: for "loop interactions" it is one of multiple types of loops (hairpin loop, internal/bulge loop, multiloop), while for "external interaction" it is outside of any RNA loops. Hence RBPs that recognize their target sites mainly

by their sequence content can vary in their preference for these different structural environments of ssRNA. There are also other RBPs that recognize their target sites mainly by their shape and geometry and not by their sequence content. These proteins usually bind double-stranded RNA. For example, proteins with double-stranded RNA-binding domains (dsRBDs) bind stems of dsRNA with at least 10 base pairs and the interactions are with the helix backbone, rather than with specific bases.

RBPs dynamically coordinate the fates of mRNAs in response to various conditions [122], and any mutation or disruption of their functions can lead to disease. For example, Fragile X syndrome is caused by the loss of function of the RBP FMR1 by the expansion of a CGG repeat above 200 units in the 5'UTR (untranslated region) [111]. The involvement of RBPs in many disease networks is presented in more details in a general review [92].

2.2 Detecting protein-DNA interactions

Identification of the specific DNA-TF interactions is vital for understanding transcriptional regulation. We focus on high-throughput experimental approaches that assess the binding of TFs either *in vitro* or *in vivo*. Each category has its own advantages and disadvantages:

1. *in vitro* methods, like universal protein binding microarray (uPBM), assess the binding specificity and affinity for a specific TF, but the TF-DNA interactions are isolated in non-biological conditions and thus they cannot capture the influence of nucleosomes or other competing/cooperating proteins;
2. *in vivo* assays, like chromatin-immunoprecipitation followed by sequencing (ChIP-seq), detect the binding sites of a specific protein in a specific cellular context, but they require TF-specific antibodies, are restricted in the cell types that they can query binding, and often times capture indirect binding events.

2.2.1 PBM experiments

A typical protein binding microarray (PBM) experiment uses a double-stranded DNA microarray on which an epitope tagged DNA-binding protein of interest is applied. A fluorophore-conjugated antibody, specific to the TF epitope tag, is then used to provide a readout of the protein-bound microarray. Both the binding sequences and the amount of the protein can be identified using the fluorescence intensities.

In order to determine high-resolution *in vitro* DNA binding specificity data for any TF and from any organism, a specific type of PBM experiments are used: universal

protein binding microarrays [11], which were created in Martha Bulyk's lab, at Harvard. The universalPBM arrays are specifically designed to contain all possible DNA 10-mer sequences in a compact, overlapping manner. Each strand of DNA is 60 nucleotides long and is attached to the glass substrate at their 3' end. The variable part of each DNA sequence consists of the last 35 nucleodites, so it contains 26 distinct 10-mers. The specific sequences used on the microarray are determined from a de Bruijn sequence of order 10. The design ensures that every 8-mer occurs at least 16 times on the array, while the non-palindromic ones occur at least 32 times.

PBM data for a large set of TFs is currently available in the UNIPROBE database [127] (see Section 4.1.1). The main advantage of this type of data is that it encompasses the binding affinities of a TF in a comprehensive manner and also that it includes the nucleotide interdependencies. One drawback of the PBM data is derived from it's *in vitro* setting: the DNA binding specificity observed in such an experiment may differ from the *in vivo* specificity of the respective TF.

2.2.2 ChIP-seq experiments

Chromatin-immunoprecipitation followed by sequencing (ChIP-seq) [62] has been extensively used for analyzing *in vivo* binding of many TFs, chromatin modifications, and other chromatin associated proteins in a variety of organisms, including human cells and tissues.

The ENCODE Consortium has performed the largest number of ChIP-seq experiments on the human genome [34] (see Section 4.1.2).

Summarily, a chromatin immunoprecipitation (ChIP) experiment consists of enriching for genomic locations to which a DNA binding factor was bound in the living cell, using an antibody specific to the protein of interest. First the proteins are cross-linked to their genomic locations and then the DNA is sheared by sonication, the cross-linked proteins remaining attached to the small fragments of genomic DNA to which they were interacting. In the next step, the protein of interest is purified and everything else is discarded. After preparing the DNA, the enriched DNA sequences are identified and quantified by high-throughput next generation sequencing. The number of sequenced reads, i.e. sequencing depth, may vary greatly between experiments, from hundreds of thousands to hundreds of millions of 25- to 75-bp sequences (called short reads). For ENCODE ChIP-seq experiments, the minimum threshold is set to 20 million mapped reads, but an experiment can reach 100 million mapped reads.

In the case of a typical transcription factor, the number of sequenced reads is positively correlated with the number of identified ChIP-seq positive sites [34]. But the peak counts typically reach saturation at particular levels of sequencing depth, depending on the specific factor binding properties [77]. Thus increasing the sequencing depth does not always lead to better results. It must be taken into account the fact that by sequencing at greater depths, many of the additional identified peaks may correspond to low-affinity binding sites and/or open chromatin regions with low specificity of TFs binding.

2.3 Detecting protein-RNA interactions

Identification of the specific RNA-RBP interactions is vital for understanding post-transcriptional regulation. For this reason, multiple low- and high-throughput experimental methods have been developed to assess the binding of RBPs either *in vitro* or *in vivo*. Each category has its own advantages and disadvantages:

1. *in vitro* methods are able to offer a complete characterization of the sequences and/or structures that an RBP can, or cannot bind, but the RBP-RNA interactions are isolated in non-biological conditions and their power largely depends on the design of complex RNA libraries;
2. *in vivo* assays have the potential to identify the genome-wide targets of the RBPs, but require RBP-specific antibodies, and are restricted in the cell types that they can query binding.

2.3.1 SELEX experiments

SELEX (Systematic Evolution of Ligands by EXponential enrichment) is a low-throughput procedure for *in vitro* identification of TF or RBP binding preferences [33]. A randomized RNA oligonucleotide pool is used to select high-affinity binding sequences in several sequential cycles of binding to purified protein and amplification by polymerase chain reaction (PCR). After that the final set of short sequences preferred by the RBP are identified and analyzed in order to assess the binding specificity. We note that, because of its design, the SELEX assay will only detect the highest affinity RNA target sites. Therefore one drawback is that the range and relative affinity of RNA-sequence preferences of an RBP are not completely characterized.

Recently, a high-throughput version of this method was developed, often called HT-SELEX (high-throughput SELEX) [157]. This is a more quantitative and comprehensive procedure in which the number of binding rounds is reduced to a few or even to a

single one, but millions of RNA oligos are sequenced. Thus it overcomes the disadvantage of SELEX, producing a more quantitative estimate of the RBP binding specificity. HT-SELEX was also use in the context of DNA-protein interactions, and the binding preferences for hundreds of TFs were estimated with this type of assay [63].

2.3.2 RNAcompete experiments

RNAcompete is an *in vitro* method related to HT-SELEX, which uses a smaller, designed, initial RNA oligo pool instead of the large and complex SELEX initial pool [120]. The oligo pool is synthesized by means of a custom-designed microarray that ensures the presence of approximately 244,000 short RNAs, between 30 and 38 nucleotides (nt) long. The design of the microarray is similar to that of the universal PBM experiments [11], being based on modified de Bruijn sequences. In order to obtain an unbiased measurement of the relative sequence-binding preferences of RBPs, the microarray template ensures that each 7nt RNA sequence appears either in ssRNA or weakly paired RNA in at least 128 oligos. After the RNA library is generated, a single pull-down of the RNAs bound to the tagged RBP of interest is performed. Then the relative abundances of each oligo is measured using a microarray with the same format as for the pool generation. The small pool size and the use of a custom-designed microarray for assessing the results make RNAcompete much less expensive than HT-SELEX. One downside is the inability to capture strict structural requirements on the RBPs binding sites due to the absence of RNAs with stable secondary structure in the oligo pool. However, RNAcompete was used to investigate the RNA sequence preferences for more than 200 RBPs, and the results are reported in CisBP-RNA database (`http://cisbp-rna.ccbr.utoronto.ca/`) [120] (see Section 4.2.1).

2.3.3 CLIP-based experiments

Cross-Linking and ImmunoPrecipitation (CLIP) is a large-scale *in vivo* type of assay. The standard CLIP method [145] involves the use of ultraviolet (UV) light to induce permanent cross-links between RNAs and RBPs *in vivo*, which prevent re-association of protein-RNA complexes *in vitro* or other nonspecific pull-downs. After the photo-crosslinking, the cell is lysed and the target RBP is isolated using immunoprecipitation. Then the bound RNA frangments are partially digested, so only the sequence in direct contact with the RBPs is obtained. Thus this type of assay has high resolution when determining the actual sites of RNA-RBP interaction. Due to its high quality and precision results, CLIP is a state-of-the-art technology [75].

Some variants of the CLIP method have been developed, all of which are associated with high-throughput sequencing and are currently used: HITS-CLIP, PAR-CLIP and iCLIP. High-throughput Sequencing CLIP (HITS-CLIP or CLIP-seq) [24] combines high-throughput sequencing with the standard CLIP procedure. Individual-nucleotide resolution ultraviolet Cross-Linking and ImmunoPrecipitation (iCLIP) [76] pinpoints protein-RNA crosslinked nucleotides during sample preparation. In Photo-Activatable Ribonucleoside enhanced CLIP (PAR-CLIP) [53], living cells are cultured with a photoreactive ribonucleoside analog, such as 4-thiouridine (4-SU) or 6-thioguanosine (6-SG), to facilitate cross-linking. This leads to 100- to 1,000-fold higher cross-linked RNA recovery, and also to UV radiation-induced mutations that highlight the cross-linked sites. These can be T-to-C mutations in the case of 4-SU (most common case) or G-to-A mutations for 6-SG, and are used to improve the identification of the RBP footprint.

RBPs binding sites identified by CLIP variant experiments are reported in databases like doRiNA (`http://dorina.mdc-berlin.de/`) [3] and CLIPZ (`http://www.clipz.unibas.ch/`) [70].

Chapter 3

Background in bioinformatics

This chapter provides some bioinformatics concepts and methods that lay at the foundation of this thesis. I start with brief descriptions of the methods used to process the raw experimental data, and follow with computational tools that predict the RNA secondary structure. Then I present available methods for the identification of RNA-binding proteins motifs and finish the chapter with a selection of basic machine learning techniques.

3.1 Processing experimental data

3.1.1 Processing microarray assays

There are multiple experimental assays based on microarrays, a technology for the simultaneous analysis of thousands of samples on a solid substrate [131]. A DNA microarray or DNA chip is a collection of microscopic DNA spots attached to a glass surface, each spot containing a specific DNA sequence, called probe or oligo, that has 25-60 bases, depending on design. Certain biological properties can be measured by using a population of fluorescent-labeled target RNA or DNA molecules and detecting the probe-target hybridization. The microarray read-out consists in one or multiple images containing the laser-scanned probes intensities, at one or more wavelengths (see Fig. 3.1). The processing of microarray data involves two main aspects: 1) image processing and 2) data correction and normalization. Briefly, the image processing can be divided into the following steps: grid alignment, segmentation into foreground and background grid spots, foreground and background quantification (mean, median, interquartile range, threshold, etc.) and spot quality assessment. The resulted data is then preprocess using a number of approaches such as background correction, logarithm transformation and

different types of normalization. After this, the fluorescence intensities of all microarray probes are reported and further analysis is necessary to obtain biologically meaningful results [37].

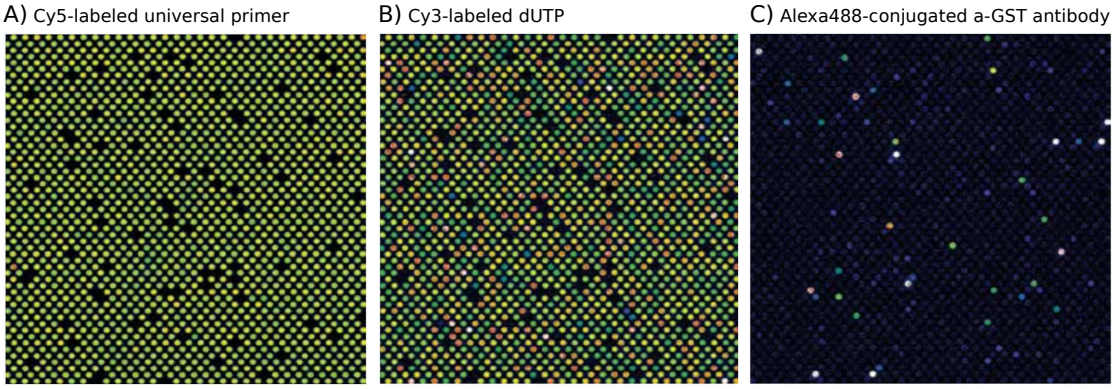


FIGURE 3.1: Visualizing microarray data: zoom-in images for the same PBM microarray, scanned at different wavelengths in each of the three main stages of an experiment: (A) Cy5-labeled universal primer for primer annealing, (B) Cy3-labeled dUTP for primer extension and (C) Alexa488-conjugated a-GST antibody for protein binding step. Fluorescence intensities are shown in false color, with blue corresponding to low signal intensity, green to moderate signal intensity, yellow to high signal intensity, and white to saturated signal intensity. The blank spots are single-stranded negative control probes. Figure adapted from [11] (Figure 1).

In this work, we use data from two assays that employ microarrays:

- universal PBM, for protein-DNA interactions (see Section 2.2.1).
- RNAcompete, for protein-RNA interactions (see Section 2.3.2).

To accurately estimate the capacity of a given protein to bind to a specific sequence, it is better to use shorter sequences, that will be present multiple times on the array, and to consider the median intensity of these occurrences. In the case of universal PBM experiments, the microarrays are designed so that every 8-mer occurs at least 16 times, while the non-palindromic ones occur at least 32 times. Therefore, the data from a universalPBM experiment can be used to determine the binding preference of the given protein to all possible contiguos 8-mers [11]. This approach avoids possible experimental biases towards specific spots on the microarray, or towards specific position of the 8-mer on the DNA strand (closer to the loose end or to the substrate). Beside the 8-mers median intensities, the binding specificity can be expressed also in the form of 8-mers enrichment scores (E-scores) or of 8-mers Z-scores. The E-scores are more robust than the intensities to changes in protein concentrations and experimental conditions. They range from -0.5 to 0.5, and higher values correspond to higher sequence preference. Usually an E-score value above 0.35 is considered to correspond to a binding event. In the case of RNAcompete experiments, the design ensures that each 8-mer is represented at least 12 times in the unstructured probes, and data is used to compute the median intensities and E-/Z-scores for all possible contiguos 7-mers in a simmilar manner [120].

3.1.2 Processing genome-wide sequencing assays

Currently there are many types and variants of assays that use high-throughput sequencing to detect specific aspects of cell biology. They all share the same idea of measuring a certain biological property by purifying a population of hundreds of millions of DNA molecules. We primarily use two assays that employ genome-wide sequencing:

- ChIP-seq, for protein-DNA interactions. In this case the sequenced DNA molecules correspond to regions bound by the specific protein of interest (see Section 2.2.2).
- CLIP methods, for protein-RNA interactions. In this case the sequenced DNA molecules result from reverse transcription of RNA molecules bound by a protein of interest (see Section 2.3.3).

The sequences obtained with the massively-parallel sequencing platforms are called "reads" or "raw reads". They need to be aligned back to the genome and then the "aligned reads" are used to determine enriched locations, called "peaks", in a genome-wide manner.

3.1.2.1 Read alignment

The first step in processing high-throughput sequencing data consists in aligning (mapping) the generated reads to a reference genome. It is not a trivial task and can be accomplished with one of the almost 100 available software tools [43]. The sequencing reads are relatively short fragments of DNA sequences (in the order of tens of nucleotides) and it is a challenge to find their true location in the genome given that they can contain both technical sequencing errors and genetic variation. Therefore, mapping tools take into account not only exact matches, but also locations with one, two or more base mismatches. Another uncertainty arises when a given read matches multiple locations in the genome. These reads can be treated in different ways: either discarded altogether or filtered based on a score that combines the number of genomic locations with mismatch information. In the case of assays like RNA-seq, reads need to be aligned to transcripts and thus transcript splicing has to be considered because a read can be split in the reference genome if it covers the end of one exon and the beginning of another.

The choice of alignment algorithm and of parameters used (number of mismatches, fixed or variable read length, use or elimination of reads that map to multiple positions) will impact the resultant set of reads. In the case of DNA data, popular read alignment tools include BWA [85], Bowtie [80] and Bowtie2 [79], while for RNA data commonly used mappers are RSEM [84], STAR [29] and Tophat [71].

3.1.2.2 Finding enriched regions

After the reads are mapped, a common processing step involves determining enriched locations in genome-wide sequencing data. These regions are called "peaks" and the task of identifying them is often called "peak calling". There are different tools that accomplish it and they use diverse strategies to search for regions in the genome with increased sequence reads density above background.

Reads generated by different types of experimental protocols have specific read coverage profiles. For example ChIP-seq reads tend to be quite spread out compared to the very specific PAR-CLIP reads (see track 4 vs. 7 in Fig. 3.2). Because of this many peak-callers are designed for a specific type of experimental data. Furthermore, some tools even use a certain property of a specific experimental protocol to find peaks, as is the case of PARalyzer [21], a tool designed for PAR-CLIP reads that scores potential RNA-protein interaction sites taking into consideration the locations of the diagnostic mutations. In the example from Fig. 3.2, track 5 depicts the 5 peaks recovered by PARalyzer. Each peak can be associated with a score that reflects the binding evidence or the binding strength, peak p5.1 being the most enriched in this window. For ChIP-seq data, recommended peak callers include MACS [156], SPP and JAMM [61]. The resulted peaks can vary in terms of number, size and quality. In the example from Fig. 3.2, the ENCODE peak p2.1 is really long with more than 2,600 bases, while the corresponding custom peak p1.1 has around 1,400 nucleotides. The shorter peaks p2.2 and p2.3 have little read coverage and are not present in the custom set, designed to have high confidence peaks.

3.2 Predicting RNA structure

RNA molecules consist in a single and flexible strand of nucleotides that can fold into multiple stable conformations. Their secondary structure is influenced by mutiple factors: the bases they contain, the local environment, and interactions with other molecules. Therefore it is difficult to obtain the exact secondary structure that an RNA has during an interaction with a protein *in vivo*. All *in silico* predictions take into account just the sequence of nucleotides. Furthermore, it is computationaly infeasible to predict the structure of a hole RNA molecule if that RNA is relatively long. Most structure prediction tools are able to fold stretches of RNA up to a few hundred bases long. The available approaches are based either on free energy minimization [12, 159], or on ensembles of secondary structures [28, 128]. The more recent algorithms focus on local conformations and can take into account multiple suboptimal structures.

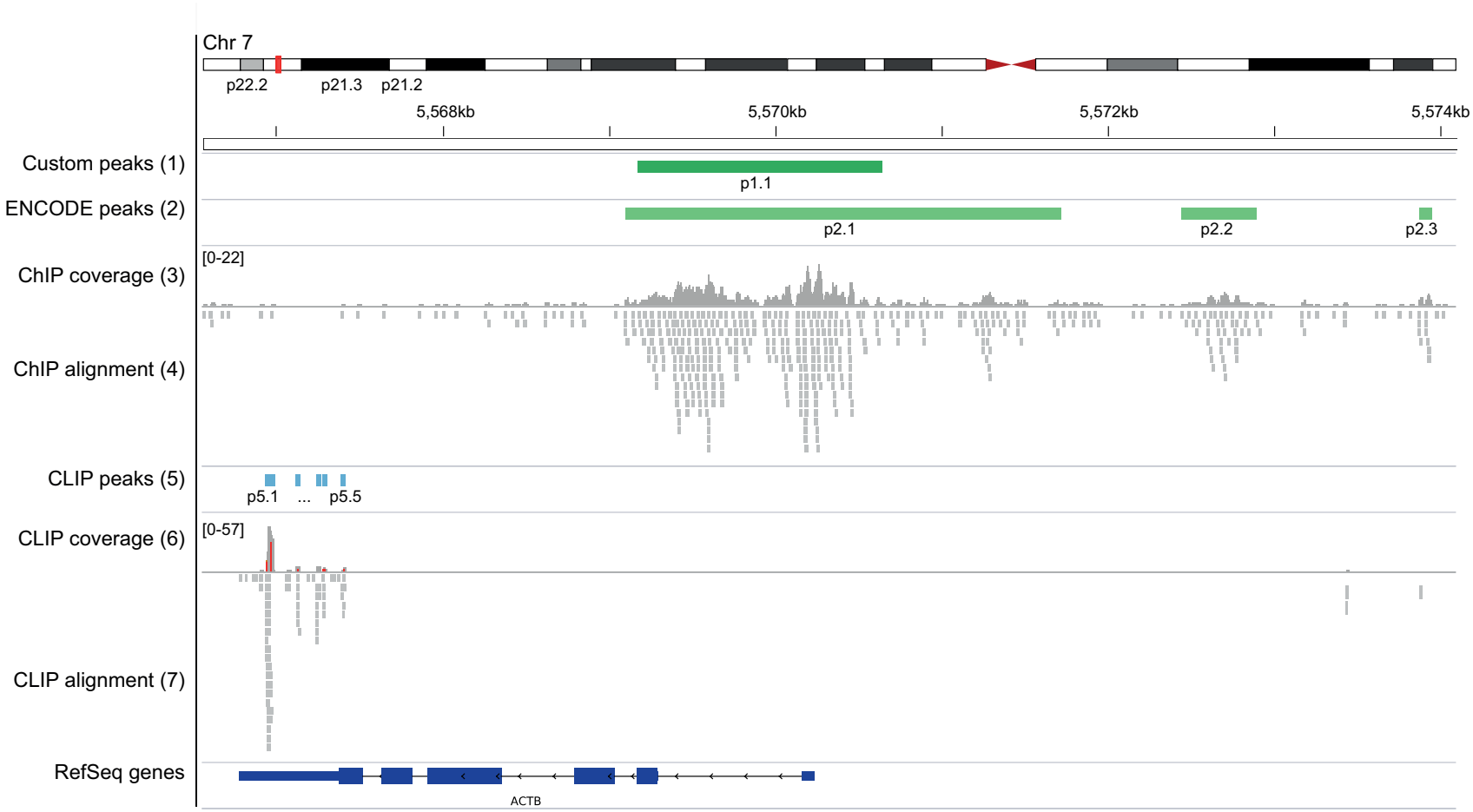


FIGURE 3.2: Visualizing high-throughput sequencing data: IGV browser snapshot of ChIP and CLIP data containing the ACTB gene locus. Tracks 1-4 correspond to ChIP-seq data for c-Myc protein performed in K562 cells and tracks 5-7 represent PAR-CLIP data for HuR/ELAVL1 in HEK293 cells. For the ChIP-seq experiment, we depict the aligned reads (4) and their coverage (3) together with two sets of peaks: one set reported by the authors, derived with PeakSeq1.0 (2) and a different set derived with a custom procedure that involves MACS, SPP and IDR (see Section 6.2.8.1) (1). For PAR-CLIP we show the aligned reads (7), their coverage (6) and the peaks obtained with PARalyzer (5). Hans-Hermann Wessel (Max Delbrück Center, Berlin, Germany) assisted with the figure generation.

3.2.1 Tools based on free energy minimization

The first RNA secondary structure predictions based on energy minimization using nearest neighbor energy parameters were made in the early 1970s [25, 144]. In less than a decade were developed multiple efficient folding algorithms that used dynamic programming to optimize the thermodynamic free energy [110, 146, 160]. The early folding methods were able to compute a single minimum energy structure of an RNA sequence. Also, the dynamic programming algorithms consider only structures with fully nested base pairs, ignoring pseudoknots.

The Mfold software was developed by Zuker in the late 1980s [158]. The "M" refers to "multiple" and the algorithm predicts not only a minimum free energy folding, but also minimum free energy structures conditional on containing specific base pairs. The software improved over the years and multiple Mfold versions were released [159]. The RNA folding model incorporates thermodinamic parameters from the Turner group [96].

RNAplfold is a more recent tool that predicts RNA single-strandedness using free energy minimization [12]. It is a software from the ViennaRNA package that uses locally stable secondary structures. RNAplfold considers all possible overlapping windows of W nucleotides over the RNA sequence of interest and derives the minimum free energy structure for each of them. Then for each nucleotide, the tool considers all windows that contain it and outputs the average base pair probability.

3.2.2 Tools based on ensembles of secondary structures

Since an RNA molecule may adopt multiple structures in the cell, another approach to folding is to consider an ensemble of probable structures. The first statistical sampling algorithm for structure prediction, Sfold, was developed by Ding and Lawrence in the early 2000s [28]. It uses the Boltzmann equilibrium probability of a secondary structure I for a given sequence S:

$$P(I) = \frac{e^{\frac{-E(S,I)}{RT}}}{U} \tag{3.1}$$

where $E(S, I)$ represents the free energy of the structure I for the sequence S, R is the gas constant, T is the absolute temperature and U is the partition function for all secondary structures of S, $U = \sum_I e^{\frac{-E(S,I)}{RT}}$. Sfold applies a sampling procedure that uses Boltzmann probabilities and then clusters the 1000 sampled structures into classes. The minimum free energy structure has the largest sampling probability and thus is present in the ensemble. For each class of similar structures, it considers as representative the structure that has the highest frequency.

A newer ensemble-based method is RNAprofiling [128]. Like Sfold, RNAprofiling uses a statistical sample of 1000 RNA secondary structures from the Boltzmann ensemble of possible RNA secondary structures associated with a given RNA sequence. Instead of clustering the ensemble, the tool focuses on the arrangement of helices at the substructure level and reports the most frequent double-stranded regions. In this manner, RNAprofiling identifies and reports local dominant helices.

3.3 Identifying the target sites of RBPs

Computational analysis of RBP-RNA interactions is vital for interpreting the experimental data and finally understanding how an RBP finds and binds to its targets. Finding sequence motifs is not trivial due to the shortness of the binding motif and the large number of input sequences that can include many false positives. Incorporating secondary structure preferences into motif models adds an extra layer of challenges due to the noisiness of RNA structure prediction and the need of a reliable model for sequence-structure motifs that is also easy to interpret. The RBP binding motifs can be derived either with methods developed for DNA-binding proteins, which consider only the RNA primary sequence, or with specifically designed tools that account for different levels of secondary structure information (Table 3.1).

3.3.1 Tools based on sequence

MatrixREDUCE was originally design for TF-DNA interactions and is based on fitting a statistical mechanical model [40]. One distinctive feature is that MatrixREDUCE takes as input quantitative values associated with each sequence in the dataset rather than predefined “bound” or “unbound” sets.

MEME (multiple expectation maximization for motif elicitation) [8] is a very popular motif discovery algorithm which was designed originally to find sequence patterns in DNA or proteins. REFINE (relative filtering by nucleotide enrichment) [126] is an extension of MEME which first removes regions of target sequences that are relatively depleted of discriminatory hexamers and then applies MEME.

cERMIT (conserved Evidence Ranked Motif Identification Tool) [46] considers the rank-order of genome-wide binding sites and can be used both for DNA and RNA sequences. It models the binding motif as a regular expression (consensus sequence) over a degenerate alphabet and optimizes the motif score either based on a rank-order statistic or based on a regression model [21]. Since an exhaustive search over the space of all potential motifs is

TABLE 3.1: Motif finding algorithms used to analyse RBP-RNA interactions

Motif finder	Input	Strategy	Output (motif)	Structure
MatrixREDUCE [40]	Sequences and scores	Least-square fit	PWM	No
MEME [8]	Positive (and negative) sequences	EM	PWM	No
REFINE [126]	Positive sequences	extends MEME	N/A	No
cERMIT [46]	Sequences and scores	Rank order statistics	PWM	No
MEMERIS [58]	Positive and negative sequences	extends MEME	PWM	Single-strandedness guides sequence motif finding
StructRED [41]	Positive and negative sequences	extends MatrixREDUCE	PWM in a hairpin loop	Hairpin loops up to 7 bases with at least 3 paired bases
CMfinder [154], RNApromo [119]	Positive sequences	SCFG & EM	structured sequence	Sequence and structure modeled together by CM-based motif model
#ATS [88]	Positive and negative sequences	expected number of accessible target sites	IUPAC	Sequence and accessibility modeled together
RNAcontext [66]	Sequences and (affinity) scores	Limited-memory BFGS	PWM with structural context scores	Sequence and structure modeled separately
Zagros [7]	Positive sequences	extends MEME	PWM with pairedness scores	Sequence and pairdness modeled together
GraphProt [97]	Positive and negative sequences	SVM with graph-kernel	graph-based sequence and structure motifs	Sequence and structure modeled together as hypergraph

computationally infeasible, cERMIT uses a greedy search strategy. The heuristic search uses as seed motifs all possible 5 or 6-mers and evolves each of them independently until the motif score stops improving. Candidate motifs are obtained at each iteration by shorthening, lengthening, or increasing/decreasing the sequence degeneracy at one position in the motif. After convergence, all evolved motifs are clustered and the top-scoring candidates are reported.

3.3.2 Tools based on sequence and structure

The secondary structure of RNA can either be computationally predicted or experimentally measured. There are tree main approaches to account for structural information when predicting binding motifs:

- restricting the motif finding to preferred secondary structure contexts (MEMERIS [58], StructRED [41] and #ATS [88])
- using stochastic context-free grammars (SCFG) to model secondary structure (covariance models [31], CMfinder [154] and RNApromo [119])
- using other approaches to directly model sequence and structure binding sites (RNAcontext [66], Zagros [7] and GraphProt [97])

3.3.2.1 Methods with defined structural context

The first motif finders designed for RBPs used RNA secondary structure as prior knowledge to restrict the search for sequence motifs to either single-stranded regions or to specific loop structures.

MEMERIS [58] is an extension of the popular DNA motif finding tool MEME, designed specificaly for RNA. MEMERIS considers that proteins do not bind dsRNA, but only ssRNA for which the structural conformation can be either a hairpin loop, an internal loop, or the single-stranded sequences between two stems. Thus MEMERIS measures the single-strandedness of a substring in a given RNA sequence, by precomputing for each k-mer its probability of being in single-stranded context. These values are then used as priors on possible motif start positions so the search for motifs favors the single-stranded regions.

StructRED (structural cis-regulatory element detector) [41] detects RNA cis-regulatory elements that are located in hairpin loops and extends MatrixREDUCE [40]. Briefly, StructRED is a data integrative, regression-based algorithm that investigates only putative small stem-loop structures in order to find secondary structure-defined cis-regulatory elements (SCREs). StructRED uses computationally predicted stems of length three,

that can form through Watson-Crick base pairs, and loops of lengths three to six nucleotides, without considering the thermodynamic stability of the stem in the naked mRNA. This limited representation of secondary structure elements makes it difficult to apply StructRED to RBPs that do not bind hairpin loops or other types of ssRNA.

#ATS (expected number of accessible target sites) [88] is a discriminative motif-finding method that incorporates RNA accessibility. This approach fits a degenerate consensus sequence motif model which distinguishes between sets of bound and unbound transcripts, using a score equal to the sum of the accessibilities of sites in the transcript. #ATS builds the motif using a greedy heuristic that uses as seeds the top five 6-mers with respect to their predictive power, and iteratively refines them until the discriminative power of the motif stops improving. The refinement steps consist in shortening, lengthening, or introducing degenerate bases in the motif. After convergence, only the top-scoring motif is selected. We note that the use of #ATS is limited to only single-stranded binding sites, because of the way it models RNA accessibility.

3.3.2.2 Methods based on stochastic context-free grammars

Covariance models (CMs) [31] were first introduced by Eddy & Durbin in 1994 and model both the primary sequence and the secondary structure consensus of an RNA. They are probabilistic RNA motif models that generalize hidden Markov models (HMMs), being in the same time specialized stochastic context free grammars (SCFG). CMs are trained through an EM iteration loop between aligning individual sequences to a single CM and refining the CM based on the alignment. Their performance depends on the availability of a good initial alignment to seed the search. CMs are implemented in the software package COVE [31] and are largely used to define RNA families (e.g., [52] for families of ncRNAs). They are not very well suited for identification of RBP binding sites because of the short length of a typical binding site (between 2 to 10 nucleotides).

CMfinder [154] and RNApromo [119] are both fitting CM-based motif models and are using strategies based on thermodynamic stability to initialize the structural alignment of potential RBP binding sites. Establishing this initial alignment required for the CM iterations is difficult due to the fact that, unlike many RNA families, sets of RBP target sites usually do not have conserved sequence in paired regions. For initialization, CMfinder uses shared secondary structures among the minimum free energy structures of the input sequences, while RNApromo uses nonredundant substructures that are overrepresented in the positive set versus the background set.

Even if they are able to model complex RNA structures, CMs tend to overpredict base pairing and search preferentially for motifs presented in paired regions (stems). Thus they poorly represent a large set of RBPs that bind unpaired, single-stranded RNA.

3.3.2.3 Methods with sequence and structure motifs

RNAcontext [66] is a discriminative motif finding method designed to detect the relative preferences of an RBP for multiple structural contexts. The motif modeled by RNAcontext has two components: the sequence preferences in the form of a PWM (position weight matrix) over the nucleotide alphabet {A, C, G, T}, and the relative structural preferences at nucleotide-resolution. The structure preferences are defined with one of the following alphabets [65]:

- {P, H, I, M, E}, where P = paired, H = hairpin loop, I = internal loop, M = multiloop, and E = external loop
- {P, H, T, E}, where P = paired, H = hairpin loop, T = internal loop or multiloop, and E = external loop
- {P, L, E}, where P = paired, L = hairpin, internal or multi- loop, E = external loop
- {P, U}, where P = paired, U = unpaired

The secondary structure preferences are estimated using Sfold [28] or RNAplfold [12]. The context annotation of each nucleotide is defined as the probability of that base to be in a particular structural context and takes the form of a distribution of all possible contexts (similar to a PWM). This model takes into account the fact that a given RNA sequence can have multiple, distinct stable secondary structures. RNAcontext has been applied to *in vitro* binding-affinity data (RNAcompete-derived datasets). Although it was also applied to CLIP datasets, its performance in this case is not established.

GraphProt [97] uses a graph approach to simultaneously model the primary sequence and the secondary structure of RNA, extending the representation utilized by GraphClust [57]. Briefly, an RBP-bound site is modeled by a hypergraph that consist of:

(ground level) a directed graph that models the relations between nucleotides. The vertices correspond to nucleotides, and the edges to bonds between them (the sequence backbone and the structure base-pairs)

(abstract level) structure annotations for small overlapping subgraphs and the relations between them. The following posible substructures are used: stem (S), multiloop (M), hairpin (H), internal loop (I), bulge (B), and external region (E).

GraphProt uses RNAshapes [137] to sample the population of all possible structures and retains multiple representative candidates. The graph kernel used by GraphProt is the Neighborhood Subgraph Pairwise Distance kernel (NSPD Kernel). This kernel is extended in multiple ways: to allow for the hypergraph model instead of simple graphs, to work with a directed graphs instead of undirected graphs, and to consider only local structures (called viewpoints) instead of the global one.

GraphProt fits their predictive model using either Support Vector Machine (SVM) classifiers or Support Vector Regression (SVR), depending on the available information. The predicted bound model is hard to interpret or visualize, and the tool outputs the top-scoring 1000 sequences and structures, that can be converted to PWMs or logos. GraphProt was tested on both CLIP-seq and RNAcompete datasets, with high classification/regression performance.

Zagros [7] is an extension of the MEME algorithm designed for CLIP data. It performs *de novo* motif discovery, using as input only the binding sites derived from experimental data. Zagros uses a probabilistic model to account for cross-linking modification events and/or for secondary structure in the form of paired-unpaired probabilities. The structure is predicted with a modified version of RNAFold from the ViennaRNA package [90]. We note that, unlike in GraphProt setting, this predictions take into account only the relatively short peaks derived from CLIP experiments and the paired-unpaired probabilities may not reflect the local folding of the RNA molecule. Zagros extends the classic mixture model with two orthogonal components: one that includes information about crosslinking using the prior on motif occurrence indicators and another that includes information about secondary structure by expanding the sequence component of the model. The model is fit with the expectation maximization (EM) algorithm.

These motif finders are designed specifically for a certain type of experimental data and only Zagros finds *de novo* sequence-structure motifs, while RNAcontext and GraphProt work in classification or regression settings.

3.4 Machine learning techniques

A key role in many areas of science, finance and industry is played by tools and techniques to learn from data. Depending on data availability, the machine learning algorithms can be supervised, unsupervised, or semi-supervised.

The supervised learning scenario consists in having some input variables ($x_1, x_2, \ldots, x_n \in X$) with corresponding output values ($y_1, y_2, \ldots, y_n \in Y$) and using an algorithm to learn

the mapping function f from the input space to the output space:

$$Y = f(X) \tag{3.2}$$

The goal is to approximate the mapping function so well that it is possible to predict the output value y_i associated with new input data x_i. The input variables are also called "features" and represent quantitative or categorical information about a given set of objects (such as genes, people or proteins). The data used to build the prediction model is called training data. This type of learning is called "supervised" because of the known values for the outcome variable that guide the learning process. Depending on the type of the output variable, supervised learning problems can be grouped into:

- classification, if the output variable is categorical, such as "disease" and "no disease" or "bound" and "not bound". In this case the outcome categories are called "classes".
- regression, if the output variable is continuous, such as "gene expression" or "binding strength".

Unsupervised learning corresponds to only having input data $(x_1, x_2, \ldots, x_n \in X)$ and no output variables. The goal in this case is to model the underlying structure or distribution in the data. There are no labels to guide the learning, and the unsupervised learning problems can be grouped into:

- clustering, if the purpose is to group similar data together. In this case the algorithms need to use an appropriate similarity metric, and the number of clusters that are suitable/best describe the data need to be determined.
- association, if the goal is to discover association rules that describe large portions of the data.

Semi-supervised learning problems occur when a large amount of input data $(x_1, x_2, \ldots, x_n \in X)$ is available, but only some of it is labeled with the output values $(y_1, y_2, \ldots, y_k \in Y, k < n)$. In this case a mixture of supervised and unsupervised techniques can be used iteratively in an attempt to use all available information and obtain better results than supervised or unsupervised learning on only parts of the data. The semi-supervised learning can be grouped into:

- transductive learning, if the goal is to infer the correct labels for the given unlabeled data $(y_{k+1}, y_{k+2}, \ldots, y_n)$.
- inductive learning, if the purpose is to infer the correct mapping from X to Y.

In the following we focus on a selection of methods and procedures associated with classification. We first describe two state-of-the-art classification algorithms: support vector

machines and random forest and then provide some methods and metrics regarding performance estimation.

3.4.1 Classifiers

3.4.1.1 Support vector machines

A widely used supervised classification algorithm is represented by the support vector machines (SVMs) developed by Vapnik in the mid '90s [22]. Intuitively, the idea is to find an optimal hyperplane that separates between a given set of data points which belong to one of two classes (Fig. 3.3). The hyperplane searched by SVMs is the one that ensures the largest minimum distance to the training examples. In the SVM theory, this distance is called "margin" and the points closest to the separating hyperplane are called "support vectors". In other words, the goal of an SVM classifier is to maximize the margin of the training data, thus minimizing the risk of misclassifying unseen test samples.

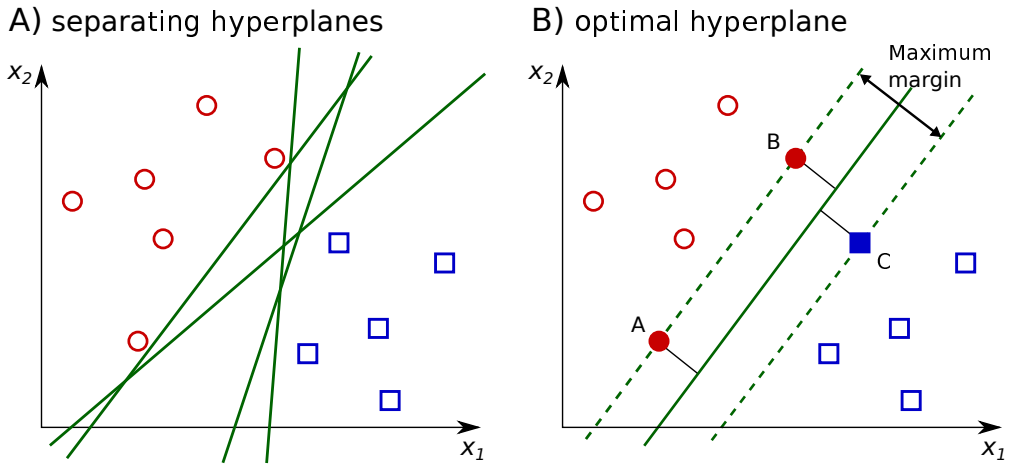


FIGURE 3.3: Illustration of support vector machines with a simple example of two classes with linearly separable data points in the two-dimensional space. There are multiple hyperplanes (in this case straight lines) that can separate the two classes (A), but the optimal line is the one that passes as far as possible from all points (B). The separating boundary is determined by the points A, B and C, called support vectors.

Support vector hyperplanes. Fig. 3.3 illustrates the case when the two classes are linearly separable. Formally, let X be a set of points $x_i \in \Re^d$ with $i = 1, \ldots, n$. Each point x_i has a label $y_i \in \{-1, +1\}$, belonging to one of two possible classes. The set X is linearly separable if there are $w \in \Re^d$ and $w_0 \in \Re$ such that

$$y_i(w \cdot x_i + w_0) \geq 1, i = 1, \ldots, n \tag{3.3}$$

The pair (w, w_0) defines the hyperplane of equation $w \cdot x + w_0 = 0$, named the separating hyperplane. The signed distance of a point x_i to the separating hyperplane (w, w_0) is given by $d_i = \frac{w \cdot x_i + w_0}{||w||}$. From the linear separability it follows that $y_i d_i \geq \frac{1}{||w||}$, therefore $\frac{1}{||w||}$ is the lower bound on the distance between points x_i and the separating hyperplane (w, w_0). The optimal separating hyperplane is then the separating hyperplane for which the distance to the closest (either positive or negative) points in X is maximum, therefore it maximizes the margin thickness $\frac{1}{||w||}$. The optimization problem can be defined as:

$$\min_{w,w_0} \frac{1}{2}||w||^2 \tag{3.4}$$

$$\text{subject to } y_i(w \cdot x_i + w_0) \geq 1, i = 1, \ldots, n \tag{3.5}$$

This is a convex optimization problem (quadratic criterion with linear inequality constraints) with $d + 1$ parameters ($w \in \Re^d$ and $w_0 \in \Re$). In the following we describe a quadratic programming solution using Lagrange multipliers. The Lagrange (primal) function, to be minimized with respect to w and w_0, is:

$$L_P(w, w_0, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i (y_i(w \cdot x_i + w_0) - 1) \tag{3.6}$$

with $\alpha_i \geq 0, i = 1, \ldots, n$. Finding the minimum implies setting the derivatives to 0:

$$\frac{\partial L_P}{\partial w_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.7}$$

$$\frac{\partial L_P}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \implies w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{3.8}$$

By substituting these constraints into (3.6) we obtain the so-called Wolfe dual:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{3.9}$$

The dual optimization problem is defined by:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \tag{3.10}$$

$$\text{subject to } \sum_{i=1}^{n} y_i \alpha_i = 0$$

$$\alpha_i \geq 0, i = 1, \ldots, n$$

This is a simpler convex optimization problem, and with its optimal solutions $\bar{\alpha_i}$ we can compute the optimal solution $(\bar{w}, \bar{w_0})$ of the primal problem (3.4) by using:

$$\bar{w} = \sum_{i=1}^{n} \bar{\alpha_i} y_i x_i \tag{3.11}$$

$$\bar{\alpha_i}(y_i(\bar{w} \cdot x_i + \bar{w_0}) - 1) = 0 \text{ for any } i = 1, \ldots, n \tag{3.12}$$

From (3.12) we can see that:

- if $\alpha_i > 0$, then $(y_i(w \cdot x_i + w_0) = 1$ which means that x_i is on the margin boundary;
- if $(y_i(w \cdot x_i + w_0) > 1$, in other words if x_i is not on the margin boundary, then $\alpha_i = 0$.

Therefore most α_i are null and the vector w is a linear combination of a relative small percentage of the points x_i. These points are called support points or support vectors because they are the closest points to the optimal separating hyperplane and the only points of X needed to determine it.

The optimal separating hyperplane produces a function $f(x) = \bar{w} \cdot x + \bar{w_0}$ and the problem of classifying a new data point x is simply solved by looking at $sign(f(x))$. We note that, by construction, none of the training data points fall in the margin area, but this will not necessarily be the case for test points. This linear SVM classifier is based on the idea that a large margin on the training points will lead to good separation on the test observations.

Support vector classifiers. In the more common case when the two classes are not linearly separable, a possible generalization is to allow for some points to be on the wrong side of the margin. Therefore we can introduce n non-negative "slack" variables $\xi = (\xi_1, \xi_2, \ldots, \xi_n)$ such that the constraints from (3.5) are replaced by:

$$y_i(w \cdot x_i + w_0) \geq 1 - \xi_i, \forall i = 1, \ldots, n \tag{3.13}$$

The generalized optimal separating hyperplane is the solution to the following optimization problem:

$$\min_{w, w_0} \left(\frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i \right) \tag{3.14}$$

$$\text{subject to } y_i(w \cdot x_i + w_0) \geq 1 - \xi_i, \forall i = 1, \ldots, n \tag{3.15}$$

$$\xi_i \geq 0, \forall i = 1, \ldots, n \tag{3.16}$$

In this formulation, the value ξ_i represents the proportional amount by which the prediction $f(x_i) = w \cdot x_i + w_0$ is on the wrong side of its margin. Misclassification occurs when $\xi_i > 1$, so bounding the sum $\sum_{i=1}^{n} \xi_i$, bounds the total proportional amount by which predictions fall on the wrong side of their margin.

The Lagrange (primal) function is in this case:

$$L_P(w, w_0, \alpha) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(w \cdot x_i + w_0) - (1 - \xi_i)] - \sum_{i=1}^{n}\mu_i\xi_i \quad (3.17)$$

This function is minimized with respect to w, w_0 and ξ_i and the associated dual optimization problem is:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i \cdot x_j \quad (3.18)\\ \text{subject to } &\sum_{i=1}^{n} y_i\alpha_i = 0 \\ &0 \le \alpha_i \le C, i = 1, \ldots, n \end{aligned}$$

Similarly, we have:

$$\bar{w} = \sum_{i=1}^{n}\bar{\alpha}_i y_i x_i \quad (3.19)$$

$$\bar{\alpha}_i(y_i(\bar{w} \cdot x_i + \bar{w_0}) - 1 + \bar{\xi}_i) = 0 \quad (3.20)$$

$$(C - \bar{\alpha}_i)\bar{\xi}_i = 0 \quad (3.21)$$

with nonzero coefficients $\bar{\alpha}_i$ only for those observations x_i for which the constraints in (3.13) are exactly met, due to (3.20). These points x_i are called as before the support vectors, since $\bar{w}$ is represented in terms of them alone. These support points will fall into one of the following categories:

- observations that have $\bar{\xi}_i = 0$ and lie on the edge of the margin. They are characterized by $0 < \bar{\alpha}_i < C$.
- points that have $\bar{\xi}_i > 0$ and $\bar{\alpha}_i = C$.

As before, the optimal separating hyperplane produces the function:

$$f(x) = \bar{w} \cdot x + \bar{w_0} \quad (3.22)$$

and the corresponding classifier is represented by $sign(f(x))$, called the support vector classifier. Sometimes referred to as linear SVM with soft margins, this classifier will give weights to all misclassified points, no matter how far away from the decision boundary,

the results being tunned by the cost parameter C. A large value for C focuses attention more on (correctly classified) points near the separating hyperplane and thus minimizes the number of misclassified points. A small C value involves data further away and maximizes the margin $\frac{1}{||w||}$. The support vector hyperplane defined for separable classes corresponds to $C = \infty$.

Support vector machines. The two support vector classifiers described so far are both linear methods that find linear decision boundaries in the input space. They can become more flexible by expanding the feature space $\Re^d$ to a higher dimensional space $\Re^m (m > d)$. In general, linear boundaries in the enlarged space translate to non-linear boundaries in the original space and thus achieve better class separation (Fig. 3.4). These enhanced non-linear classifiers are called support voctor machines (SVM).

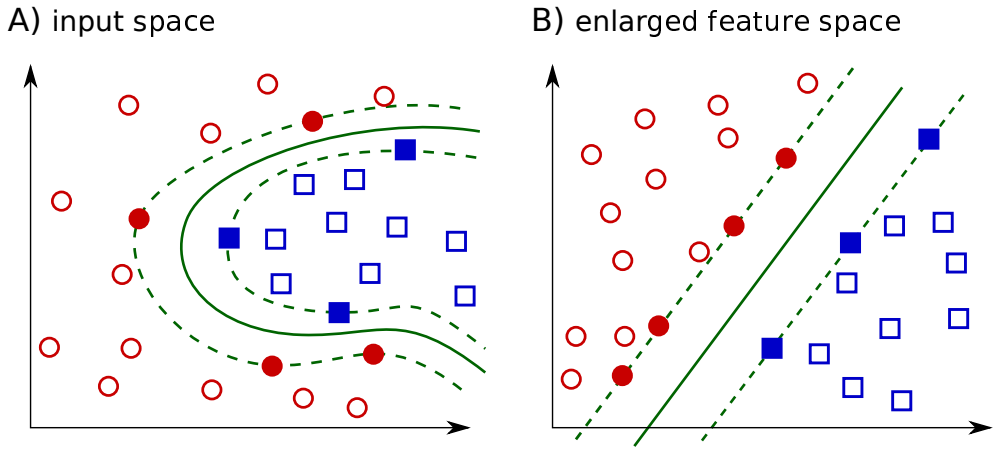


FIGURE 3.4: Illustration of a support vector machine with a polynomial kernel. The non-linear decision boundary in the input feature space (A) is obtained from a separating hyperplane in the enlarged feature space (B). The support vectors are indicated by solid fill.

Briefly, let Φ be the mapping function $\Phi : \Re^d \rightarrow \Re^m$. The support vector classifier can be fit using as features: $\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_n)$. The optimal separating hyperplane produces in this case a non-linear function $f(x) = \bar{w} \cdot \Phi(x) + \bar{w_0}$, and the classifier is, as before, $sign(f(x))$. The optimization problem (3.18) and its solution (3.22) have an interesting property: they have a form in which the input data points appear only via inner products. The corresponding Lagrange dual function is:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \tag{3.23}$$

Using (3.19), the solution function $f(x)$ can be rewritten to:

$$f(x) = \sum_{i=1}^{n} \bar{\alpha_i} y_i \Phi(x_i) \cdot \Phi(x) + \bar{w_0} \tag{3.24}$$

The non-linear SVMs are based on the observation that we do not need to specify the transformation $\Phi(x)$ if we use a so-called "kernel" function K that computes inner products in the transformed space:

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{3.25}$$

Using this type of function, it is possible to compute the separating hyperplane without explicitly carrying out the map into the higher space.

Some popular kernel functions for the SVM classifiers are:

- the polynomial (of degree q): $K(x_i, x_j) = (x_i \cdot x_j + c)^q$
- the radial basis function (RBF): $K(x_i, x_j) = e^{\frac{||x_i - x_j||^2}{2\sigma^2}}$
- the sigmoid: $K(x_i, x_j) = \tanh(ax_i \cdot x_j - b)$

The linear SVM can be viewed as a special case where we use the linear kernel function: $K(x_i, x_j) = x_i \cdot x_j$.

In the enlarged feature space perfect separation between the classes is often achievable and so the role of the cost parameter C is more straight-forward. A large value of C will prevent any positive ξ_i, determining a more wobbly (and possible overfit) boundary in the input space, while a small C will favor a small value of $||w||$, causing the boundary to be smoother.

3.4.1.2 Random forests

Another widely used classifier is random forest (RF), developed by Breiman in 2001 [14]. It has a very different approach, being an ensemble of multiple classification trees. In order to present the RF classifier, we will first describe decision trees and ensemble classifiers.

Decision trees. One of the most intuitive classifiers is the decision tree. It is a discriminative classifier that iteratively splits training data into groups and evaluates the "homogeneity" within each group, splitting again if necessary. A decision tree partitions the feature space into a set of rectangles and then associates a class label to each one (Fig. 3.5). The internal nodes of a tree correspond to classification features, the edges

correspond to possible values for the feature of the parent node, while leafs represent the classification outcome. For example, in the case of the discrete variable "eye color", a node in the tree consists in the variable name and the branches correspond to the 4 possible values {blue, brown, gray, green}. In the case of a continuous variable, like "height", a node represents a possible split of the values, like "height > 1.6", and the two edges correspond to "true" and "false" values.

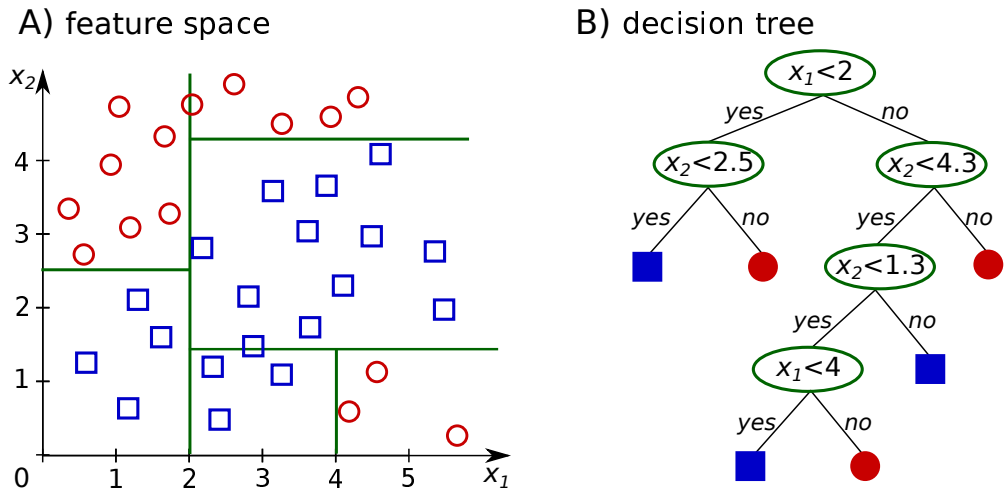


FIGURE 3.5: Illustration of decision trees in a two-dimensional feature space. Panel (A) shows the partition of the input space by recursive binary splitting and panel (B) presents the corresponding decision tree.

Decision trees are constructed with top-down induction algorithms that choose at each child-note the variable that best splits the set of associated data points. The basic outline is presented in Algorithm 1. In the example from Fig. 3.5, the input set X consists in 34 data points in $\Re^2$ and the corresponding label set contains 15 instances of class 1 (red circles) and 19 instances of class -1 (blue squares). The first split is at $x_1 = 2$, and the root note of the tree is $x_1 < 2$. Then the region $x_1 < 2$ is split at $x_2 = 2.5$ and the region $x_1 \geq 2$ is split at $x_2 = 4.3$. Only one of the branches corresponding to the $x_2 < 4.3$ node ends in a leaf node, associated in this case with class 1, so the process continues with two more splits at $x_2 = 1.3$ and then at $x_1 = 4$, until all points in one rectangle region are from the same class. This recursive partitioning is a greedy algorithm used by most decision tree learners. The main difference between different algorithms comes from the metrics used to estimate the BESTFEATURE for each split. The most popular methods and their measure for the subsets homogeneity are:

- iterative dichotomiser 3 (ID3) with information gain [117]
- C4.5 and C5.0 (successors of ID3) with information gain [118]
- classification and regression tree (CART) with Gini impurity [15]

Briefly, the information gain is based on the Shannon entropy from information theory. It can be viewed as the expected reduction of the entropy of the instance set X due to

Algorithm 1 Building a decision tree

Require: X a set of n data points points $x_1, x_2, \ldots, x_n$ over the feature space $A = \{a_1, a_2, \ldots, a_p\}$ and Y a set of corresponding class labels $y_1, y_2, \ldots, y_n$.
Ensure: Decision tree D.

```
function BuildTree(X, A)
    if (A = ∅) or (y_i = c, ∀x_i ∈ X) then
        D.AddNode(most common class in {y_i | x_i ∈ X})
    else
        a = BestFeature(X, A)
        D.AddNode(a)
        for v ∈ Values(a) do
            D.AddBranch(BuildTree(X(a = v), A \ {a}))
        end for
    end if
end function
```

a splitting on the attribute a:

$$IG(X, a) = H(X) - \sum_{v \in Values(a)} \frac{|X(a = v)|}{|X|} H(X(a = v)) \tag{3.26}$$

where $H(X)$ represents the entropy of X, $H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$, and $X(a = v)$ represents the subset of points in X for which $a = v$.

Gini impurity represents the probability that a random data point x_i that corresponds to a given node is incorrectly labeled if it is randomly labeled using the label distribution of the points in the node. The minimum value (zero) is reached when all data points associated with a node have the same class label. If the subset of observations $\{x_i\}$ in node N have k possible class labels $c_1, c_2, \ldots, c_k$, then the Gini impurity (Gini index) is defined as:

$$i(N) = \sum_{j=1}^{k} p(c_j)(1 - p(c_j)) = 1 - \sum_{j=1}^{k} p(c_j)^2 \tag{3.27}$$

The CART algorithm selects the splitt attribute a for a parent node N using the decrease in impurity:

$$\Delta i(N, a) = i(N) - \sum_{v \in Values(a)} \frac{|X(a = v)|}{|X|} i(N(a = v)) \tag{3.28}$$

where $N(a = v)$ represents the child node of N that corresponds to the $a = v$ branch, and X denotes, as before, the set of instances in N.

Ensemble classifiers The results obtained with individual classifiers can be improved using ensemble methods [26]. An ensemble of classifiers is a learning algorithm that constructs a set of classifiers and then combines their predictions in order to achieve

better classification performance. A new data point is classified by the ensemble with a weighted vote procedure.

The most common type of ensemble methods train the same type of classifier on different subsets of the available training data. The idea is that instead of learning a single (weak) classifier, learn many (weak) classifiers that are good at different parts of the input space. Another possibility is to create an ensemble that contains the same type of classifier, but trains multiple models with different parameters. There are also ensemble methods that use different classification algorithms.

Some well-known ensemble learning algorithms include bagging and boosting, both methods using a single learning method on different training data sets. Bagging refers to bootstrap aggregating and is a committee-based approaches [13]. Given a training set X of size n, bagging uses bootstrapping to generate m new training subsets X_i of equal size n', with $n' \leq n$. This means that X_i are randomly sampled from X with replacement, using an uniform distribution, and thus they may contain duplicate data points. Then m classification models are trained independently on the m bootstrap training samples and their votes are combined in order to classify new data.

Like bagging, boosting is also a committee-based approaches that trains multiple classifiers of the same type, the most popular boosting algorithm being AdaBoost (Adaptive Boosting) [45]. AdaBoost is adaptive in the sense that it learns the classifiers iteratively and after each iteration it modifies the training data, increasing the importance of the instances that have not been correctly classified. Given a training set X of size n, AdaBoost applies weigths $w_1, w_2, \ldots, w_n$ to each of the training instances $x_i, i = 1, .., n$. For the first iteration all weights are equal, $w_i = \frac{1}{n}$, and the first model is trained as usual. Then the weights of the misclassified points are increased, while the weigth for the correctly classified instances are decreased and the next classifier is trained on this data. At each iteration, the current classifier is forced to focus on the data points that were misclassified by previous models. The prediction of the ensemble is obtained with a weighted vote of all the successive classifiers.

Random forests Random forests (RFs) or random decision forests are ensemble classifiers based on decision trees. They extend the idea of a bagging algorithm for decision trees with an additional layer of randomness. Therefore each tree is trained on a bootstrap sample of the input data. In addition, each split is determined by a random subset of the available features in a process referred to as "feature bagging". The two levels of randomness are introduced in order to reduce the correlation between trees in the forest. The classification output is represented by the majority vote in the RF.

A RF has two parameters: *ntree*, the number of trees in the forest, and *mtry*, the number of variables in the random subset at each node. Depending on the size and type of the training data, a few hundred to thousands of trees are used, the value for *ntree* being usually optimized in order to reduce the training error. In classification, the typical value for *mtry* is $(\sqrt{p})$, where p is the number of total features.

The design of RFs induces two interesting properties:

- Cross validation is unnecessary, since the test error can be estimated with the **Out-of-bag (OOB) error**. This refers to the observed error of the out of bootstrap (OOB) samples for each corresponding tree. Thus the OOB error is computed from the mean prediction error on each training sample x_i, using only the trees that trained on bootstrap samples that did not include x_i.
- The ensemble classifier can be used not only for prediction, but also to assess the **variable importance**. There are two different metrics for the importance of features that can be derived from a RF classifier, namely Gini importance and permutation importance.

Given the set of features $A = \{a_1, a_2, \ldots, a_p\}$ on which a RF was trained, the Gini importance of a feature a_i represents the total decrease in Gini impurity of all nodes in the forest, where a_i was selected for splitting:

$$\Delta I(a_i) = \sum_{N \in Nodes(a_i)} \Delta i(N, a_i) \tag{3.29}$$

The Gini importance has been shown to be biased when variables have different types, favoring continuous variables and features with many categories [139]. A less biased Gini importance can be obtained by sampling training data without replacement.

The permutation importance of a feature a_i represents the average decrease in classification accuracy when the values of the respective variable are randomly permuted:

$$PI(a_i) = \frac{1}{ntree} \sum_{t=1}^{ntree} [A_{OOB}(t) - A_{OOB}^{(a_i)}(t)] \tag{3.30}$$

where $A_{OOB}(t)$ represents the prediction accuracy for tree t using OOB samples, while $A_{OOB}^{(a_i)}(t)$ represents also the OOB accuracy for tree t, but after randomly permuting the values of a_i. The variable importance described by Eq. 3.30 is known as the unscaled/raw permutation importance. The scaled permutation importance, or z-score, is obtained by dividing this average decrease of accuracy by its standard error over all trees in the RF:

$$z_i = \frac{PI(a_i)}{\sqrt{\frac{\sigma^2}{ntree}}} \tag{3.31}$$

Random forests have become increasingly popular because are robust against overfitting and can deal with "small n large p"-problems or correlated predictor variables.

3.4.2 Performance evaluation

The performance of a classifier or predictor is usually assessed with a cross-validation (CV) technique. The goal of cross-validation is to test the model on an independent data set. A classifier is trained on a dataset of known data, called the training dataset, and then it is tested against a dataset of first seen observations, called the testing dataset. The most common type of cross-validation is the k-fold cross-validation, in which the input dataset is randomly partitioned into k equal sized subsets. This cross-validation procedure consists in k steps (folds), in which a single subset is extracted from the data and kept for testing, while the rest of $k-1$ subsets are used to train the classifier. The results from the k folds can be then averaged or aggregated into a single estimation. The advantage of this method is that it uses all observations for both training and testing, while each observation is used for testing exactly once. The parameter k can depend on the size of the input dataset, with 10-fold CV being commonly used. The k-fold CV is a non-exhaustive cross-validation method that is difficult to apply on small datasets. In these cases leave-one-out CV (LOOCV), an exhaustive cross-validation technique is more appropriate. For a dataset of size n, LOOCV consists in n steps in which a single observation is left out for testing and the classifier is trained on the rest.

There are multiple metrics that can be used to measure the performance of a classifier or predictor. In the case of binary classification, members of one class are often denoted "positives" (P), while members of the other class are called "negatives" (N). To evaluate a classifier, its predictions are compared to the known classes, resulting in the following 4 categories of observations:

- true positives (TP) – positive samples that were correctly classified,
- true negatives (TN) – negative samples that were correctly classified,
- false positives (FP) – negative samples that were considered to be positives,
- false negatives (FN) – positive samples that were misclassified.

The classification accuracy (acc) is defined as:

$$acc = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.32}$$

and measures the ratio of all samples that are correctly categorized. The values range between 0 and 1, with 1 meaning perfect prediction, 0 meaning perfect anti-prediction, and 0.5 denoting random prediction. This simple statistic depends on the prevalence of the data (the proportion of negative and positive samples).

Two fundamental prevalence-independent statistics are sensitivity (sn) and specificity (sp), defined as:

$$sn = \frac{TP}{P} = \frac{TP}{TP + FN} \text{ and } sp = \frac{TN}{N} = \frac{TN}{FP + TN} \tag{3.33}$$

The sensitivity (or true positive rate, or recall) shows how many of all positives were identified. It can be seen as the probability that the prediction is positive given that the sample is negative. The specificity (or true negative rate) represents how many of all negatives were identified. As with sensitivity, it can be considered the probability that the prediction is negative given that the observation is positive. In practice, sensitivity and specificity are inversely proportional to one another to some extent.

In addition to sensitivity and specificity, the performance of a binary classification test can be measured with precision (pr) or positive predictive value. It is defined as:

$$pr = \frac{TP}{TP + FP} \tag{3.34}$$

and shows how many of the positive predictions were correctly classified.

For a more global performance evaluation, the receiver operating characteristics (ROC) is used. The ROC curve is created by plotting the true positive rate (sensitivity) on the y-axis versus the false positive rate ($1-$specificity) on the x-axis and depicts the relative trade-offs between true positive (benefits) and false positive (costs). The area under the ROC curve (AUC) is used as a summary statistic and is equivalent to the Wilcoxon test of ranks. Random classification results in a ROC represented by a diagonal line between $(0, 0)$ and $(1, 1)$ and an AUC of 0.5, while perfect classification corresponds to the $x = 0, y = 1$ scenario with an AUC of 1.

Chapter 4

Data sources

This chapter contains basic information about a selection of available data sets for different types of experimental data used in this thesis. First *in vitro* and *in vivo* datasets are presented for protein-DNA interactions, followed by *in vitro* and *in vivo* datasets for protein-RNA interactions. The chapter ends with a brief overview of RNA-seq datasets.

4.1 Protein-DNA data

4.1.1 In vitro datasets

One of the main *in vitro* methods to identify protein-DNA interactions is the universal protein binding microarray (uPBM) [11], described in section 2.2.1. PBM data for a large set of transcription factors (TFs) is available in the UniPROBE database (Universal PBM Resource for Oligonucleotide Binding Evaluation) [127]. In march 2017 the database contained DNA binding data for 566 proteins and complexes from a collection of 23 organisms. More than half of the database entries correspond to mouse proteins (299 TFs), and 55 correspond to human proteins (Table 4.1). Some "exotic" organisms have only one or two entries in UniPROBE, like the prokaryote *Vibrio harveyi*, the fungus *Mycosphaerella graminicola* or the parasitic Apicomplexan *Cryptosporidium parvum*.

From this database one can download both unprocessed output from the PBM array runs, or processed data in the form of position weight matrices (frequency matrices) and statistics for all possible contiguous 8-mer: enrichment scores (E-scores), median intensities and Z-scores.

TABLE 4.1: Number of TFs in UniPROBE by species. We show only the organisms that have more than 2 database entries.

Organism	Number of TFs
Mus musculus	299
Saccharomyces cerevisiae	119
Homo sapiens	55
Caenorhabditis elegans	24
Plasmodium falciparum	21
Arabidopsis thaliana	14
Drosophila melanogaster	14

4.1.2 In vivo datasets

Protein-DNA interactions are detected *in vivo* with Chromatin-immunoprecipitation followed by sequencing (ChIP-seq) [62] experiments (see Section 2.2.2). The ENCODE Consortium has performed the largest number of ChIP-seq experiments on the human genome [34], their aim being identification of genome-wide regulatory regions in different cell lines. In march 2017, the ENCODE database contained 5257 ChIP-seq experiments in human cells and tissues, from which 1908 corresponded to TFs ChIP-seq, the rest assaying histones or controls. These 1908 datasets cover 291 TFs (see Table 4.2), with a large number of the experiments being performed in the high priority tier 1 cell lines (K562, GM12878 and H1-hESC).

TABLE 4.2: Number of TFs with human ChIP-seq data provided by the ENCODE Consortium, by cell type.

Category	Cell type	Number of TFs
immortalized cell line	K562	76
	HepG2	47
	GM12878	32
	A549	28
	HEK293	14
	MCF-7	14
	HeLa-S3	12
	others	31
stem cell	H1-hESC	13
tissue		11
primary cell		7
in vitro differentiated cells		5
induced pluripotent stem cell line		1

From the ENCODE database one can download both unprocessed and processed data. In the case of ChIP-seq experiments, there are available fastq files (with the output of high-throughput sequencing), bam files (with aligned reads), bigWig files (for display), and narrowPeak or broadPeak files with the peaks called on the data.

4.2 Protein-RNA data

4.2.1 In vitro datasets

The main *in vitro* methods to identify protein-RNA interactions are the RNAcompete assay [120] and HT-SELEX [157], described in sections 2.3.2 and 2.3.1, respectively. The binding preferences derived from such experiments, for a large set of RBPs, are available in the CISBP-RNA database (Catalog of Inferred Sequence Binding Preferences of RNA binding proteins) [121]. In march 2017, CISBP-RNA contained RNA binding data derived from 207 RNAcompete experiments and from 43 SELEX experiments. Apart from the binding motifs determined from specific *in vitro* experiments, CISBP-RNA database contains a large number of inferred motifs (Table 4.3). For example, in the case of human RBPs there are 167 available binding motifs that correspond to 392 proteins, from which only 100 motifs are determined directly.

TABLE 4.3: Number of RBPs in CISBP-RNA by species. We show only the organisms that have more than 2 directly assessed motifs.

Species	Motifs	Direct	Inferred	RBPs	Kingdom
Homo sapiens	167	100	67	392	Metazoa
Drosophila melanogaster	60	52	8	224	Metazoa
Caenorhabditis elegans	28	15	13	211	Metazoa
Saccharomyces cerevisiae	9	8	1	95	Fungi
Plasmodium falciparum	8	8	0	103	Protists
Mus musculus	154	7	147	373	Metazoa
Trypanosoma brucei	13	6	7	138	Protists
Trichomonas vaginalis	5	5	0	208	Protists
Danio rerio	145	5	140	434	Metazoa
Xenopus tropicalis	103	4	99	317	Metazoa
Leishmania major	7	4	3	149	Protists
Gallus gallus	93	3	90	275	Metazoa
Physcomitrella patens	13	3	10	345	Plants

From the CISBP-RNA database one can download RNA binding preferences either in the form of position weight matrices and sequence logos, or as E-scores and Z-scores from each RNAcompete assay, for all possible contiguous 7-mers.

4.2.2 In vivo datasets

Protein-RNA interactions are detected *in vivo* with various CLIP experiments (see Section 2.3.3). Many laboratories perform CLIP-based experiments for specific proteins of interest. The identified RBPs binding sites are reported in databases like doRiNA (`http://dorina.mdc-berlin.de/`) [3] and CLIPZ (`http://www.clipz.unibas.ch/`) [70].

In order to have access to the unprocessed reads (and apply the same processing pipeline), the CLIP datasets of interest can be found in Gene Expression Omnibus [32] using the corresponding accession numbers. For example, all CLIP datasets used in Chapter 6 are summarized in Table 4.4.

Table 4.4: CLIP datasets on which **SSMART** was applied.

Protein	Reference	SRA accession number	Protocol	Cell line
FMR1	[4]	SRR527727	PAR-CLIP	HEK293
FMR1	[4]	SRR527728	PAR-CLIP	HEK293
FUS	[60]	SRS117977	PAR-CLIP	HEK293
HuR	[74]	SRR189777	PAR-CLIP	HEK293
HuR	[82]	SRR309289, SRR309290	PAR-CLIP	HeLa
HuR	[103]	SRR248532	PAR-CLIP	HEK293
IGF2BP2	[53]	SRS073139	PAR-CLIP	HEK293
IGF2BP3	[53]	SRS073140	PAR-CLIP	HEK293
LIN28A	[19]	SRR458758	CLIP-seq	A3-1
LIN28A	[19]	SRR458759	CLIP-seq	A3-1
LIN28A	[19]	SRR458760	CLIP-seq	A3-1
LIN28A	[54]	SRR764666	PAR-CLIP	HEK293
LIN28A	[151]	SRR531465	CLIP-seq	HEK293
LIN28A	[151]	SRS352780	CLIP-seq	H9
LIN28B	[51]	SRS420701	PAR-CLIP	HEK293
LIN28B	[54]	SRR764667, SRR764668, SRR764669	PAR-CLIP	HEK293
PUM2	[53]	SRS073141	PAR-CLIP	HEK293
QKI	[53]	SRS073142	PAR-CLIP	HEK293
ROQUIN	[108]	SRR857933	PAR-CLIP	HEK293
ROQUIN	[108]	SRR857934	PAR-CLIP	HEK293
SRSF1	[153]	SRR2107150	PAR-CLIP	HeLa
SRSF3	[153]	SRR2107156	PAR-CLIP	HeLa
SRSF7	[153]	SRR2107152	PAR-CLIP	HeLa
SRSF9	[153]	SRR2107153	PAR-CLIP	HeLa
ZFP36	[104]	SRR1046759	PAR-CLIP	HEK293

4.3 Other data

4.3.1 RNA-seq datasets

The type and quantity of RNA molecules in a biological sample at a given time can be captured by RNA sequencing (RNA-seq) experiments. RNA-seq uses next-generation

sequencing to measure expression across all or parts of the transcriptome. Depending on library preparation, RNA-seq experiments can capture:

- all RNA molecules (total RNA-seq)
- RNA molecules selected to have polyA tail (polyA RNA-seq)
- RNA molecules without the polyA tail (polyA depleted RNA-seq)
- RNA molecules selected to have specific size (small RNA-seq)
- specific RNA molecules (targeted RNA-seq)
- ribosome-protected mRNA fragments (ribosome profiling)
- etc.

The ENCODE Consortium is one of the major sources for transcriptomic data in human cell lines and tissues. Until march 2017, they performed a total of 882 RNA-seq experiments, almost half of them being polyA mRNA RNA-seq (Table 4.5).

Table 4.5: Number of RNA-seq experiments provided by the ENCODE Consortium.

Assay type	Number of experiments
polyA mRNA RNA-seq	417
total RNA-seq	260
small RNA-seq	173
polyA depleted RNA-seq	32

Chapter 5

Case study: differential *in vivo* DNA binding of paralogous TFs

5.1 Introduction[1]

The DNA binding site motifs of hundred of eukaryotic TFs have been determined thus far using high-throughput *in vivo* techniques such as chromatin immunoprecipitation (ChIP) coupled with microarray analysis (ChIP-chip [123]) or sequencing (ChIP-seq [62]), as well as *in vitro* assays such as protein binding microarrays (PBMs [11]). A close examination of the available TF-DNA binding motifs from databases such as UniPROBE [127], Transfac [98], and Jaspar [116] reveals that many eukaryotic TFs have highly similar DNA binding properties. This is not surprising given that most TFs are members of protein families that share a common DNA binding domain and thus have very similar sequence preferences [6]. However, it is surprising that, despite having the potential to bind the same genomic sites, individual members of TF families (*i.e.*, paralogous TFs) often function in a non-redundant manner by binding different sets of target genes and controlling different regulatory programs.

In this section we use the Myc/Max/Mad family of TFs as a model system to investigate whether the intrinsic DNA binding preferences can explain why paralogous TFs interact with different genomic sites *in vivo*.

5.1.1 Myc/Max/Mad family

Myc, Max, and Mad proteins are members of the basic helix-loop-helix leucine zipper (bHLH/Zip) family. They are associated with cancer and their functions are related

[1] This section was published as [105]

to cell proliferation, differentiation, and death. While the members of the Myc family operate as activators and promote cell growth and proliferation, being typically over-expressed in cancer cells, Mad proteins are transcriptional repressors, inhibiting cell proliferation and being under-expressed in human cancers. In order to bind to DNA, both Myc and Mad have to heterodimerize with transcription factor Max. On the other hand, Max can bind to itself, forming homodimers, and also can bind to a number of other transcription factors, including Myc and Mad. Not only that Myc and Mad play antagonistic roles in the cell and are competing with each other and with other factors for binding to Max, but they are also competing for putative DNA binding sites across the genome, their DNA binding affinities being very similar, namely for the E-box site CACGTG. The DNA preferences of proteins in Myc/Max/Mad family is substantiate by all the existing binding motifs for them in Transfac [98] (Fig. 5.1).

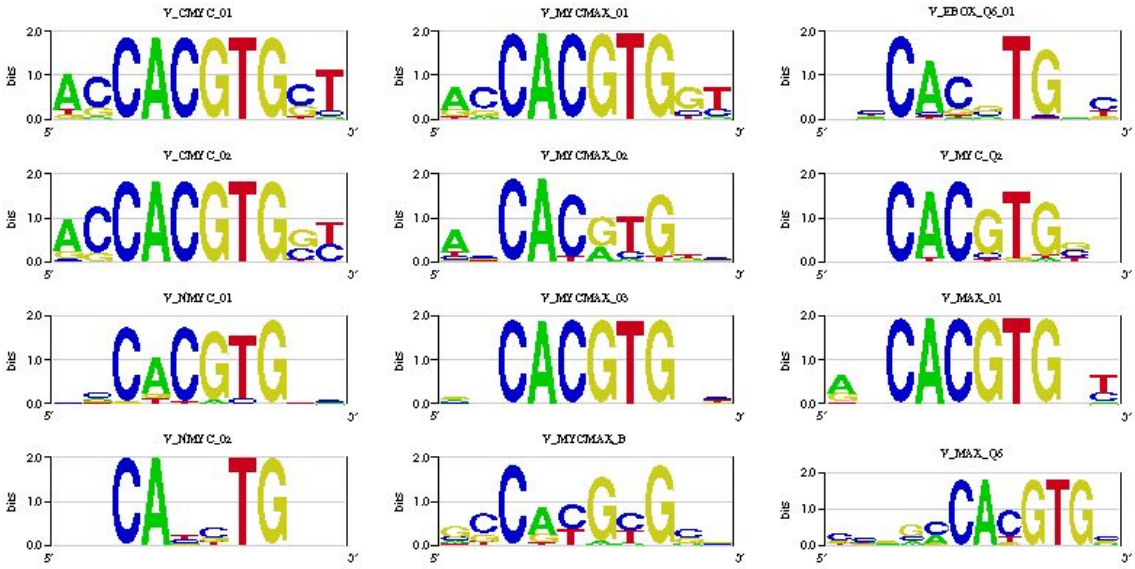


FIGURE 5.1: Various DNA binding motifs for Myc, Max, and Mad from Transfac [98]. The logos were generated using enoLOGOS[152].

5.2 Analysis[2]

5.2.1 Experimental data

In this case study we combined *in vivo* and *in vitro* TF-DNA binding data for three representative members of the Myc/Max/Mad family, namelly: c-Myc (referred to also as Myc), Mad2 (or Mxi1, referred to simply as Mad) and Max proteins. We focus our analysis on determining whether subtle differences in the intrinsic sequence preferences

[2]Part of this analysis was published as [105]

TABLE 5.1: Number of ChIP-seq peaks in HeLa S3 and K562 cell lines for Myc, Max and Mad. The third column lists the total number of peaks reported in each data set, and the last column list the number of ChIP-seq peaks that passed the ChIP-seq p-value cutoff of 10^{-10}.

Cell line	TF	# ChIP-seq peaks reported	# ChIP-seq peaks after cutoff
HeLa S3	Myc	7440	6205
	Max	11401	8232
	Mad	32138	9758
K562	Myc	3324	3162
	Max	1918	1516
	Mad	29567	6110

of Myc and Mad can explain, at least in part, the unique genomic targets bound by only one of the two factors *in vivo*.

As evidence of *in vivo* binding we used ChIP-seq data from the ENCODE project [35]. We restricted to Hela S3 and K562 cell lines because these are the only two cell lines that satisfy the following criteria:

1. ChIP-seq data is available for c-Myc, Mad2 and Max; and
2. the data sets corresponding to a specific cell line have been generated in the same laboratory.

For c-Myc, Mad2 and Max we downloaded the ChIP-seq data in narrowPeak format from the UCSC Genome Browser [129]. For the HeLa S3 cell line, 7,440 binding regions (i.e., ChIP-seq peaks) were reported for c-Myc, and 32,138 for Mad2. Because the number of bound genomic sequences varied greatly between the two TFs, it would be difficult to perform a comparative analysis directly. The fact that different types of controls were used in the c-Myc and Mad2 ChIP experiments (standard versus no primary antibody) probably contributes to the larger number of peaks reported for Mad2. However, a close examination of the ChIP-seq data also revealed that the p-value cutoffs used for reporting the peaks were very different between the two TFs: $10^{-8.8}$ for c-Myc and $10^{-2.4}$ for Mad2. To make the two data sets more comparable, we applied a cutoff of 10 for the $-\log_{10}$ of the ChIP-seq p-value. This resulted in more balanced sets of *in vivo* targets for c-Myc and Mad2, with 6205 and 9758 bound regions, respectively. A similar trend was observed for the K562 cell line, for which the same cutoff was enforced (see Table 5.1). We used these sets of targets for all the analyses described in this section.

In order to derive a background distribution of *in vivo* TF binding, we also downloaded DNase-seq data from ENCODE [35], corresponding to the two selected cell lines (HeLa S3 & K562).

We needed also high-quality DNA binding site motifs or other types of data that reflect the intrinsic DNA binding preferences of Myc/Max/Mad factors. Although such data is available for c-Myc [98, 116], none of the TFs from the Mad family have been thoroughly characterized and the only DNA motif available for Mad2 (in the Transfac database) is a general E-box motif of low quality [98]. In order to thoroughly characterize the DNA binding preferences of c-Myc and Mad2, universal PBM experiments [11] were performed in Raluca Gordân's lab. The two TFs were tested either alone or in combination with TF Max. As expected, the c-Myc:c-Myc and Mad2:Mad2 homodimers bound DNA very weakly even when tested at high concentrations, while the c-Myc:Max and Mad2:Max heterodimers bound with high affinity to many of the DNA sequences on the PBMs. PBM experiments were performed essentially as described by Berger et al. [11]. His-tagged versions of c-Myc, Mad2, and Max proteins were used in the PBM experiments, and they were a kind gift from Peter Rahl (Whitehead Institute). The proteins were expressed in E-coli and purified as described in [89]. To ensure that the DNA binding signal detected on the PBMs corresponds to the c-Myc:Max and Mad2:Max heterodimers, and not the Max:Max homodimer, the concentrations of c-Myc/Mad2 were 10 times higher than the concentration of Max. In addition, the heterodimers were tested both with tagged and untagged versions of the Max protein, and that lead essentially to the same results (see the PBM data available online at `http://www.genome.duke.edu/labs/gordan/COUGER/`). We will henceforth refer to the c-Myc:Max PBM data as c-Myc PBM data, and to the Mad2:Max PBM data as Mad PBM data.

From the universal PBM data for c-Myc and Mad2, we computed several measures of the DNA binding specificity of the two factors:

1. we used the Seed-and-Wobble algorithm [11] to derive DNA binding site motifs, or position weight matrices (PWMs) [138] (see Fig. 5.2);
2. we computed the median fluorescence intensity for each possible 8-mer, with high median intensities corresponding to 8-mers strongly preferred by the TF; and
3. we computed enrichment scores (E-scores) for each possible 8-mer. E-scores range from -0.5 to +0.5, with higher values corresponding to higher sequence preference. Typically, E-scores > 0.35 correspond to specific TF-DNA binding [11, 49].

Compared to 8-mer median intensities, the E-scores are more robust to changes in experimental conditions (e.g., binding buffers) and protein concentrations. However, 8-mer median intensities can be used to approximate the median intensities for longer k-mers, intensities that are not directly measured on the PBMs (see Subsection 5.2.3.2).

FIGURE 5.2: c-Myc, Mad2 and Max DNA binding motifs derived from *in vitro* PBM data. The logos were generated using enoLOGOS[152].

TABLE 5.2: Information about not overlapping sequences from Myc and Mad ChIP-seq data sets in HeLa S3 and K562 cell lines. The second column lists the total number of peaks in each set, and the last three columns list, for the three types of overlap, the number and percent of peaks that do not overlap.

ChIP-seq data set		Total	No physical overlap	Physical but no summit-based	Physical but no 8-mer-based
HeLaS3	Myc	6205	2786 (44.90%)	87 (1.4%)	493 (7.95%)
	Mad	9758	6308 (64.64%)	117 (1.2%)	518 (5.31%)
K562	Myc	21797	16185 (74.25%)	226 (1.03%)	-
	Mad	6110	376 (6.15%)	421 (6.89%)	-

5.2.2 Overlap analysis

Let $\mathcal{P}$ and $\mathcal{S}$ be two sets of peaks from different ChIP-seq data sets, and $P = P_1P_2 \dots P_m \in \mathcal{P}$, $S = S_1 \dots S_n \in \mathcal{S}$ two peaks of length m and n from these sets. P and S have a *physical overlap* if there exists $k > 0$ such that $P_1 \dots P_k = S_{n-k+1} \dots S_n$ or $P_{m-k+1} \dots P_m = S_1 \dots S_k$.

We define a *summit-based binding overlap* of P and S to be a physical overlap of P and S that contains both the summit of P and the summit of S.

Also, we define an *8-mer-based binding overlap* of P and S to be a physical overlap of P and S that contains 2 consecutive 8-mers with E-scores greater than 0.45 for both TFs.

We used ChIP-seq data from ENCODE [35] for Myc (c-Myc protein), Mad (Mxi1 or Mad2) and Max, in two cell-lines: HeLa S3 and K562. We are interested in comparing Myc and Mad ChIP-seq peaks, and finding the sequences that are factor-specific, ie don't have overlaps with the other set. The number of sequences that fall into this category are presented in Table 5.2. The results for HeLa S3 cell line are more balanced (Fig. 5.3). In this case, it is remarkable how many sequences are characteristic to each TF, by having no physical overlap with any sequence from the other ChIP-seq set: 45% for Myc and 65% for Mad. Also, from the sequences that do overlap physically, the percent of those who don't have a binding overlap is negligible (am average of 1.3% for summit-based overlap and 6.6% for 8-mer-based overlap).

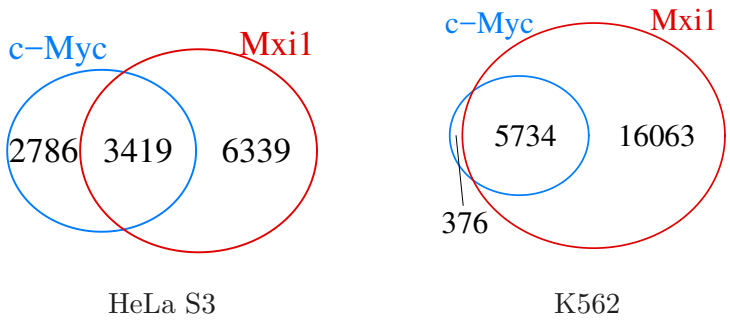


FIGURE 5.3: Physical overlaps between c-Myc and Mad2 (Mxi1) ChIP-seq peaks

5.2.3 Analysis using intrinsic DNA binding preferences

In the study of differentiation between paralogous TFs, a primary task is to analyse the intrinsic DNA binding preferences of those TFs. For this task, we used *in vitro* data from uPBM experiments, in the form of both PWM and PBM (E-scores or Intensities). We used two approaches to explore the possibility that this type of intrinsic data can explain differential *in vivo* DNA binding by c-Myc versus Mad2.

5.2.3.1 PWM data & AUC enrichment

We examined each ChIP-seq data set by computing the enrichment of various motifs in that data set. This enrichment represents the area under the receiver operating characteristic (ROC) curve (AUC). The motifs we used for this task were the three PBM-derived PWMs from Gordân lab (see Fig. 5.2), together with 12 PWMs from Transfac that are associated with the proteins from Myc/Max/Mad network (see Fig. 5.1). To compute all AUC enrichments, three steps were made for each ChIP-seq data set:

(A) We determined a negative set (background) of sequences that was associated with the positive set (foreground – all the sequences from the ChIP-seq data set). From a DNase I hypersensitive sites sequencing (DNase-Seq) data set for the specific cell-line, set that contained all genome-wide regulatory regions, we randomly chosen a set of sequences that met three conditions: none of the sequences do not overlap with any sequence from the original ChIP-seq data set, their number is equal to the number of foreground sequences, and they have the same length distribution as the positive set. Also, we compared the distributions of the GC content for the foreground and background sequences, finding that they were similar.

(B) For each sequence, positive or negative, and each of the 15 PWMs we computed the probability that the corresponding TF binds that sequence, in the following manner:

Let $S = S_1 S_2 \dots S_L$ be a DNA site of length L, and T a TF, with ϕ the corresponding motif, of length L: $\phi(b, i)$ = the probability of finding base b at position i in the binding site, where $b \in \{A, C, G, T\}$ and $1 \leq i \leq L$. Let also ϕ_0 denote the background model, obtained from all the sequences in DNase-Seq data set.
The probability that TF T binds DNA site S, of the same length as the PWM, can be written as:

$$\begin{aligned} P(T \text{ binds } S) &= \frac{[T \cdot S]}{[T \cdot S] + [S]} = \frac{[T]}{[T] + K_d(T, S)} \\ &= 1 \Big/ \left(1 + \frac{1}{[T]} \cdot \prod_{i=1}^{L} \frac{\phi_0(S_i)}{\phi(S_i, i)}\right), \end{aligned} \tag{5.1}$$

where the dissociation constant, $K_d(T, S) = [T] \cdot [S]/[T \cdot S]$, was approximated by the ratio of the scores for site S according to the PWM and background model: $K_d(T, S) = \prod_{i=1}^{L} \frac{\phi_0(S_i)}{\phi(S_i, i)}$, and the concentration of free TF, $[T]$, is set to the dissociation constant for the site with the optimal PWM score.
Then the probability that TF T binds a DNA sequence X of an arbitrary length n, with $n > L$, is:

$$\begin{aligned} &P(T \text{ binds } X) = P(T \text{ binds any } X_j \dots X_{j+L-1}) \\ &= 1 - \prod_{j=1}^{n-L+1} \left(1 - 1 \Big/ \left(1 + \frac{1}{[T]} \cdot \prod_{i=j}^{j+L-1} \frac{\phi_0(S_i)}{\phi(S_i, i-j+1)}\right)\right). \end{aligned} \tag{5.2}$$

(C) For each PWM we ranked all the sequences in descending order of the binding probability and then computed the AUC statistics, using the formula:

$$AUC = \frac{1}{B + F} \left(\frac{\rho_B}{B} - \frac{\rho_F}{F}\right) + 0.5 \,, \tag{5.3}$$

where B and F represent the sizes of the background and foreground, respectively; ρ_B and ρ_F depict the sums of the background and foreground ranks.

After computing AUCs for each ChIP-seq data set, we compared these enrichments (Fig. 5.4), keeping in mind that an AUC of 1 is associated with perfect enrichment, while an AUC of 0.5 depicts the enrichment of a random motif. Regarding the enrichments of the three PBM-derived motifs, they have the same relations in all ChIP-seq data sets, specifically Myc and Mad have almost the same AUC (0.665 and 0.663 in Myc ChIP-seq, 0.585 and 0.582 in Mad ChIP-seq), while Max has a slightly greater enrichment (0.688 in Myc ChIP-seq and 0.611 in Mad ChIP-seq). The AUC enrichments of the motifs from Transfac are generally smaller. There are even two motifs, V_NMYC_02 and V_EBOX_Q6_01, that have AUCs very close to 0.5, the random value, in Myc ChIP-seq,

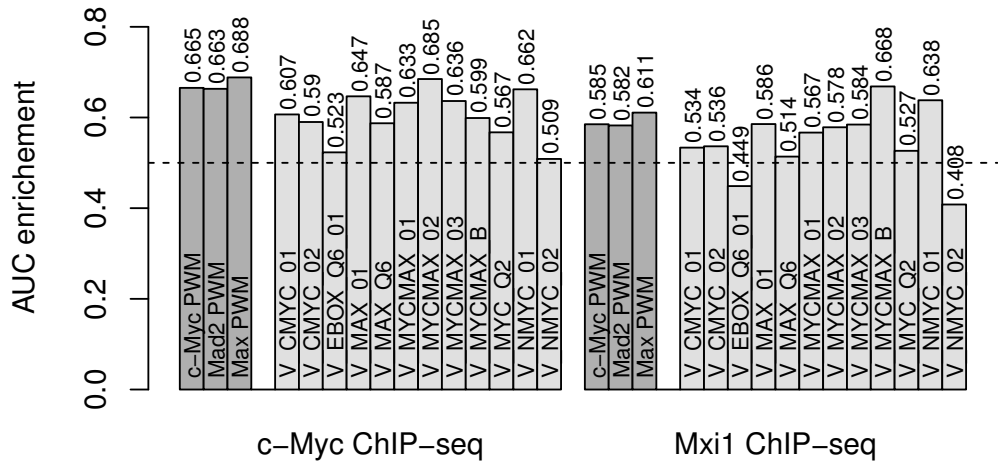


FIGURE 5.4: AUC enrichments of 15 motifs in the HelaS3 ChIP-seq data sets. Dark grey corresponds to PWM derived from *in vitro* PBM data, while light grey to motifs from TRANSFAC.

and even bellow that in Mad ChIP-seq data set. This is probably due to the fact that both motifs are more general, supporting the CANNTG binding site. In conclusion, this approach of using the motifs is unable to differentiate between the ChIP-seq data sets.

5.2.3.2 PBM data, matches and cutoffs

Another analysis we performed on each ChIP-seq data set was based on a different method of scoring them by PBM data. We considered first the 10 bp PWMs derived from universal PBM experiments for Myc and Mad and, for each TF, we obtained a list with all possible 10-mers in descending order, by their PWM score. Then, for each ChIP-seq data set and each TF we counted the number of peaks that contained at least one 10-mer in the set of 10, 100, 250, 500, 1000, 5000 and 10000 top-scoring 10-mers. We compared the fractions of sequences corresponding to Myc and Mad ChIP-seq data sets. Also, in order to compare those results with the background distribution, we computed the fractions of DNase-seq peaks containing top-scoring 10-mers for Myc and Mad. All the above fractions are reported in Fig. 5.5 (A). For both ChIP-seq data sets, the values for Myc and Mad are very similar, with the same pattern of slightly larger fractions for Mad TF, so no differentiation is possible between them. On the same time, the ChIP-seq sequences have clearly higher binding affinity than the DNase-seq peaks to either Myc or Mad. The difference between the corresponding fractions is about 0.2 for Myc data set, and 0.1 for Mad data set.

The information from the PWMs we used is in fact a synthesis of a larger and more specific collection of information, namely the E-scores from the PBM experiments. Since it is not possible to get E-scores for 10-mers from PBM data, we used 8-mer E-scores and 10-mer pseudo-intensities to perform two similar analysis, in which we computed the

fractions of sequences containing the same numbers of top-scoring k-mers, all possible 8-mers being ranked by E-score and all possible 10-mers by pseudo-intensity.

We estimated the 10-mers intensities from the 7-mer and 8-mer median intensities. Let $X = X_1X_2 \dots X_k$ be a DNA site of length k, with $I_k(X)$ it's median intensity, and $E_k(X)$ it's energy. Then the intensities for all 10-mers are computed in the following steps: compute all 7-mers and 8-mers energies from 7-mers and 8-mers intensities (Eq. 5.4); compute all 10-mers pseudo-energies using additions and subtractions (Eq. 5.5); from pseudo-energies, compute all 10-mers pseudo-intensities (Eq. 5.6).

$$E_k(X) = \log \frac{1}{I_k(X)} \ , \forall X = X_1 \dots X_k, \forall k \in \{7, 8\} \ ; \tag{5.4}$$

$$E_{10}(X_1 \dots X_{10}) = \sum_{i=1}^{3} E_8(X_i \dots X_{i+7}) - \sum_{i=2}^{3} E_7(X_i \dots X_{i+6}) \ , \forall X \ ; \tag{5.5}$$

$$I_{10}(X) = e^{-E_{10}(X)} \ , \forall X = X_1 \dots X_{10} \ . \tag{5.6}$$

Fig. 5.5 (B,C) displays the fractions of peaks containing top-scoring k-mers for pseudo-intensities and E-scores, respectively. As in the case of PWM scores, the difference between ChIP-seq data and DNase-seq data is obvious for both intensities and E-score. However, regarding the possible difference between Myc ChIP-seq data and Mad ChIP-seq data, there are some indications. The fractions corresponding to intensities show higher values for Myc in Myc ChIP-seq sequences, and also higher values for Mad in Mad ChIP-seq sequences, but only close to saturation (after 0.8), and for more than 5000 top-scoring 10-mers (ie 0.95% from the total 4^{10} 10-mers). In the case of E-scores, this difference is observable for more than 100 top-scoring 8-mers (ie 0.3% from 4^8, the total number of 8-mers). Although these results show that, regarding the task of differentiating between Myc and Mad, there is more useful information in E-scores than in PWMs, the proof is minor, uncertain in some cases, and other approaches of using this data are needed.

5.3 Discussion[3]

Most eukaryotic transcription factors are members of protein families that share a common DNA binding domain and have highly similar DNA binding preferences. However, individual TF family members (i.e. paralogous TFs) often have different functions and bind to different genomic regions *in vivo*, as observed from chromatin immunoprecipitation assays followed by microarray analysis or high-throughtput sequencing (ChIP-chip or ChIP-seq) [50, 101].

[3]Part of this section was published as [107]

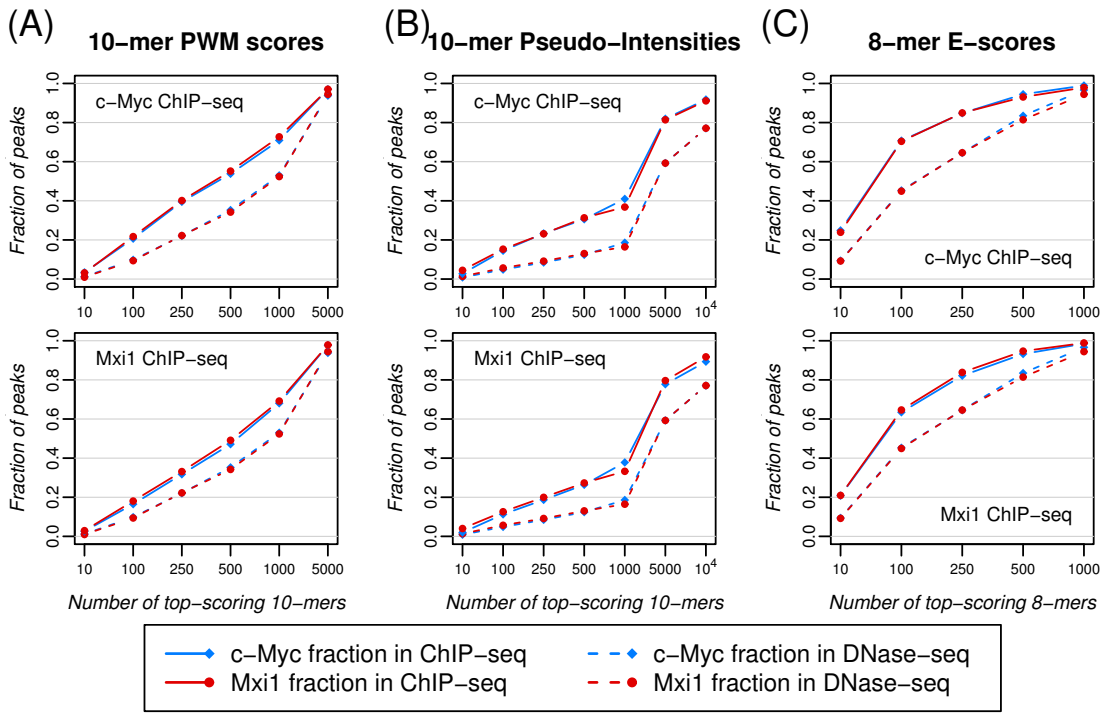


FIGURE 5.5: Myc and Mad fractions of ChIP-seq and DNase-seq peaks in HelaS3 coresponding to best k-mers, where the ranking is done with: (A) PWM score, (B) pseudo-intensities, and (C) E-score

Several mechanisms can contribute to differential *in vivo* DNA binding of paralogous TFs. First, some pairs of paralogous TFs exhibit subtle differences in DNA binding specificity—either for the core binding site [42] or for the binding site flanks [50],—and such differences can explain, at least in part, how each TF selects its unique targets. Second, paralogous TFs may interact with different protein co-factors that modulate their DNA binding specificity [135], or they may respond differently to certain chromatin environments. Third, some paralogous TFs are expressed in different cells or at different stages during cellular differentiation or during the cell cycle; in such cases, the precise chromatin environment in the cell where each paralogous TF is expressed will dictate where the TF binds in the genome.

In this chapter we focused on paralogous TFs that are present in the cell at the same time, have highly similar DNA binding specificities, but still show significant differences in their *in vivo* genomic binding profiles, as measured by ChIP-seq. Using the Myc/-Max/Mad family of TFs as a model system, we showed that the intrinsic DNA binding preferences of c-Myc and Mad2, two paralogous TFs, cannot explain why the two factors bind distinct sets of targets *in vivo*. For such paralogous TFs, interactions with different sets of protein co-factors are a likely mechanism for achieving differential *in vivo* specificity. Therefore, we designed a classification-based approach that will be the focus of the next chapter.

Chapter 6

COUGER: a tool to investigate protein-DNA interactions

6.1 Introduction

Our results from Chapter 5 show that the DNA binding motifs of two paralogous TFs, c-Myc and Mad2, cannot explain their differential *in vivo* binding. Despite the large amount of *in vivo* ChIP-seq data currently available, especially through the ENCODE project [35], computational tools for analyzing this type of differences, between the genomic binding profiles of paralogous TFs, are still lacking. In order to successfully distinguish between the genomic regions bound uniquely by such TFs, we designed a classification framework, **COUGER** (co-factors associated with uniquely-bound genomic regions), that identifies sets of putative co-factors. The rationale of this approach is that interactions with specific co-factors determines the different genomic targets of TFs with similar binding motifs.

6.2 The COUGER framework

COUGER is a general framework for identifying putative co-factors that provide specificity to paralogous TFs [107]. **COUGER** can be applied to any two sets of genomic regions bound by paralogous TFs (e.g., regions derived from ChIP-seq experiments). The framework determines the genomic targets uniquely-bound by each TF, and identifies a small set of co-factors that best explain the binding differences (Fig. 6.1).

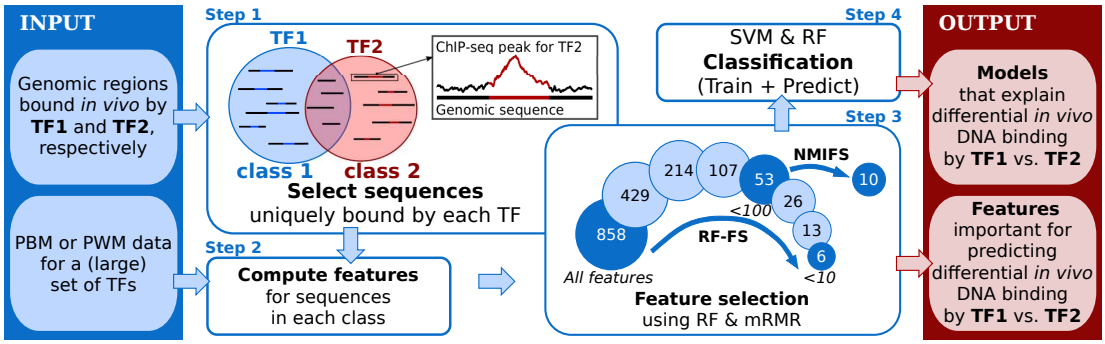


FIGURE 6.1: The **COUGER** framework. Step 1 represents the derivation of the two classes (TF1- and TF2-specific) and is omitted in the case of FASTA input files. In step 2, all features are computed for the two classes, from PBM or PWM data. Step 3 illustrates the custom feature selection procedure. The circles represent the number of features that are considered in each iteration, and the darker ones correspond to the sets of features that are used in classification. In step 4, classifiers are learned on the training set and then predictions are made on test set. Steps 3 and 4 are repeated 5 times, according to the 5-fold CV setting.

6.2.1 Classification algorithms

In our framework we use two state-of-the-art supervised classification algorithms: namely SVM (Support Vector Machine) [22] and RF (Random Forest) [14]. Some of the features in our classification task are highly correlated, and the advantage of using SVM and RF is that both algorithms are robust even in this type of environment. We use the available free software packages LIBSVM [17] and Random Jungle [132].

SVM is a widely used classifier due to its high accuracy and its successful handling of high-dimensional data. We train SVMs with both linear and radial basis function kernels ($\mathbf{SVM_{lin}}$ and $\mathbf{SVM_{rbf}}$, respectively).

RF is comparable in performance with SVM and one of its distinguishing characteristics is that it explicitly computes a measure of the importance of each variable for the classification task. This property is vital for the feature selection we are using. In this project, we apply the Random Jungle method with the unscaled permutation importance ($\mathbf{RF_{pi}}$) [132].

We apply different classifiers in order to assess the reliability of the results and their independence of particular techniques. In addition, each method has specific strengths and weaknesses ($\mathbf{SVM_{rbf}}$ usually yields better performance than $\mathbf{SVM_{lin}}$, while results obtained with $\mathbf{SVM_{lin}}$ are more interpretable).

6.2.2 Classes

COUGER performs binary classification of binding sites corresponding to two TFs

with similar binding motifs. The two classes are the DNA sequences under the ChIP-seq peaks for two paralogous TFs (TF1 and TF2), which the user can specify either in FASTA format or, for convenience, directly with ChIP-seq peak coordinates in ENCODE narrowPeak format, ENCODE broadPeak format, or BED format. TF1-specific sequences and TF2-specific sequences are defined by excluding the ChIP-seq peaks that overlap any peak of the other TF (Fig. 6.1, Step 1). In order to avoid a potential classification bias toward one of the two classes, an equal number of DNA sequences from each set is selected. Then, in the case of narrowPeak input files, which we strongly recommend, each sequence is trimmed to ±100 bp on each side of the ChIP-seq peak summit. Having the location of ChIP-seq the peaks summit, **COUGER** considers only its close vicinity because our goal is to identify co-factors that bind together with TF1 or TF2. We note that for high-quality ChIP-seq data, the TF-DNA binding events are thought to occur, in general, within 50 bp of the peak summit.

6.2.3 Features

Features reflecting the binding specificity of putative co-factors are computed from:

1. high-throughput *in vitro* TF-DNA binding data from universal protein-binding microarray (PBM) assays [11, 127], or
2. from large collections of DNA binding motifs (i.e., position weight matrices, PWMs) [63, 98] (Fig. 6.1, Step 2).

From each universal PBM data set we use the enrichment scores (E-scores) for all possible 8-mers. We note that the E-score is a modified form of the Wilcoxon-Mann Whitney statistic and ranges from -0.5 (least favored sequence) to +0.5 (most favored sequence). For a given PBM data set or PWM, and a given DNA sequence, **COUGER** uses two features:

- "MAX" or "M" (the maximum score over all the k-mers in that sequence), and
- "TOP3AVG" or "A" (the average score over the top 3 highest-scoring k-mers in that sequence – this takes into account the fact that TF binding sites may occur in clusters).

More specifically, the following steps are performed in order to compute "MAX" and "TOP3AVG" features for each specific DNA sequence (from each class):

Step 1 (PWM) A window of the same size as the PWM is slided across the sequence of interest, and a "PWM score" is computed for each window position (see Fig. 6.2). This score represents the log-likelihood ratio of that k-mer being generated by the PWM as opposed to a uniform background frequency model.

Step 1 (PBM) A window of length 8 is slided across the sequence of interest, and the "PBM score" is assigned for each window position by retrieving the corresponding 8-mer E-score.

Step 2 From all the scores derived at the first step, two numerical features are computed: the maximum score over all the k-mers in the sequence, and the average score over the top 3 highest-scoring k-mers in the sequence (see Fig. 6.2).

(A)

(B) …GCGC **TCCCCACGTGCGCGCTCACACGTGCATCGGCTACCCATGTGCCTC** GGGCT …

10-mer	PWM score
TCCCCACGTG	-7.0013
CCCCACGTGC	-23.4061
CCCACGTGCG	**7.7842**
CCACGTGCGC	-30.0300
CACGTGCGCG	-0.6539
ACGTGCGCGC	-27.9614
CGTGCGCGCT	-4.0158
GTGCGCGCTC	-40.9659
TGCGCGCTCA	-0.7386
GCGCGCTCAC	-29.8710
CGCGCTCACA	-13.2007
GCGCTCACAC	-29.5139
CGCTCACACG	-8.9880
GCTCACACGT	-31.1448
CTCACACGTG	-2.2240
TCACACGTGC	-26.7710
CACACGTGCA	**5.6353**
ACACGTGCAT	-33.8801
CACGTGCATC	-9.1585
ACGTGCATCG	-20.8069
CGTGCATCGG	-10.1774
GTGCATCGGC	-29.7659
TGCATCGGCT	-9.2720
GCATCGGCTA	-15.0440
CATCGGCTAC	-24.3472
ATCGGCTACC	-18.3647
TCGGCTACCC	-20.7391
CGGCTACCCA	-19.8933
GGCTACCCAT	-23.5238
GCTACCCATG	-17.8316
CTACCCATGT	-16.9370
TACCCATGTG	-7.4740
ACCCATGTGC	-25.4189
CCCATGTGCC	**6.7985**
CCATGTGCCT	-25.9673
CATGTGCCTC	-15.7824

FIGURE 6.2: Illustration of **COUGER**'s feature derivation for a specific PWM and a specific DNA sequence. (A) Logo of a position weight matrix (PWM), in this case for TF c-Myc. (B) The computation of "MAX" and "TOP3AVG" features for a specific DNA sequence and the c-Myc PWM. A window of size k=10 (PWM length) is slided across the sequence of interest, and a "PWM score" is computed for each window position (right column, in a heatmap manner, where green corresponds to the minimum value and red to the maximum). Next, two numerical features are derived from all the scores: the maximum score over all the k-mers in the sequence (in this case MAX = 7.7842), and the average score over the top 3 highest-scoring k-mers in the sequence – values in bold (in this case TOP3AVG = $(7.7842 + 5.6353 + 6.7985)/3 = 6.7393$).

6.2.4 Feature selection

One of the most important steps of our classification approach is feature selection, because we expect only a small number of TFs to be potential co-factors and interact with the considered paralogous TFs. **COUGER** performs feature selection using a combined procedure consisting of RF recursive feature elimination (RF-FS) and minimum Redundancy Maximum Relevance Feature Selection (mRMR) [27, 115]. This procedure is illustrated in Fig. 6.1, Step 3, which depicts the case of using 858 features derived from PBM data for 429 TFs. The combined feature selection takes into account the biological meaning of the features and of their numerical values, and its design is presented in detail in the following subsections. After the feature selection step, classification is performed on 5 feature sets:

1. all features (ALL),
2. under 100 features selected by RF-FS (RFFSu100),
3. under 10 features selected by RF-FS (RFFSu10),
4. the first 5 features selected by mRMR/NMIFS (NMIFS5), and
5. the first 10 features selected by mRMR/NMIFS (NMIFS10).

6.2.4.1 Advanced procedure for feature selection

In the first release of **COUGER**, only RF feature selection (with recursive feature elimination) was applied [105]. Random forest is one of the best classifiers to be used in a feature selection process, because it explicitly estimates the importance of each variable for the classification task. The backward elimination technique in Random Jungle (RF-FS) is an iterative process in which a random forest is grown at each step and a subset of variables is discarded. The eliminated features are those with the smallest importance. We used the same metric as for classification, namelly the unscaled permutation importance. We dropped 50% of the features at each iteration, and we stopped the algorithm when the number of features fell below 100.

One downside of RF-FS is that it does not differentiate well between correlated features and so the selected features may have a certain degree of redundancy. In the case of c-Myc vs. Mad TFs with ChIP-seq data from HeLa S3 cell line, the classification accuracy obtained with the set of features selected as described above was very good ($\approx 87\%$), but some of the features are (biologically) redundant (see Table 6.1, column "RFFSu100"). For example the factor E2F2 is present with 3 different classification features on the list of first 10 : E2F2_v1_A, E2F2_v2_A and E2F2_v2_M. (Feature names are constructed from 2 or 3 parts: TF name, version of the experiment in case of replicates, and the name of the numerical feature: M for "MAX" and A for "TOP3AVG".) The first two

represent the same type of score (TOP3AVG) on two replicate experiments, while the last two represent different **COUGER** scores for the same experimental data (denoted E2F2_v2). Moreover, other TFs from the same class have high-scoring positions on this list, namelly E2F1, E2F3 and E2F4 – that all have similar motifs. Thus it is clear that RF-FS is capable of selecting a set of relevant features, but is limited in discriminating between features derived from redundant/similar data (either closely-related TFs, or replicate experiments for the same TF, or different scores for the same experimental data).

A very large set of potential co-factors is not biologically reasonable. The number of considered features can be reduced in our setting, without a big loss in classification accuracy. When RF-FS was performed with more iterations, until less than 10 features were kept, the decrease of accuracy was of $\approx 1\%$ (see Table 6.1, column "RFFSu10"). It is notable that even if some of the selected 6 features are correlated, their power of discrimination is comparable to that of the 53 features considered previously (RFFSu100). Thus we aimed at restricting the set of features selected by RF-FS as much as possible, but in the same time to eliminate the redundant features.

6.2.4.2 Comparison between FS methods

In order to solve this task, we conducted a literature search to find suitable feature selection techniques that take into account the correlation between features. We selected and compared tree such filter methods:

- Redundancy-Based Feature selection (RBF) [155]. This procedure uses Pearson correlation coefficient for the continuous features and computes two types of correlations: Pearson correlation between each feature i with the class (individual correlation denoted $ICorr_i$), and Pearson correlation between a pair of features (i, j) with the class labels (combined correlation denoted $CCorr_{i,j}$), where the pair of features is represented as a single aggregate feature computed for example as the mean values. RBF works with the features sorted in descending order by their correlation with the class, $ICorr_i$, and in iterations over feature i and it's pairs j $(j < i)$ removes all features j that have $CCorr_{i,j} \leq ICorr_i$.
- Clearness-Based Feature Selection (CBFS) [133]. This method computes a clearness score ($CScore$) for each feature. Feature clearness is defined as the separability between classes that is induced by the respective feature, with highly clear features contributing towards obtaining high classification accuracy. $CScore$ is based on the degree of correctly clustered samples to the centroid of their class and is used to rank the features.

TABLE 6.1: Top 10 features selected by various FS algorithms, sorted by their respective score, in the case of c-Myc vs. Mad (HeLa S3 cell line). The results correspond to five FS methods and are presented on 5 different columns: RFFSu100 – under 100 features selected by RF-FS, RFFSu10 – under 10 features selected by RF-FS, RBF – redundancy-based FS, CBFS – clearness-based FS, and mRMR(2digits) – minimum redundancy maximum relevance FS with 2 digits discretization. The second row contains the obtained median accuracy by RF for the respective feature set. The names of the features are constructed from 2 or 3 parts: TF name, version of the experiment in case of replicates, and the name of the feature: M for "MAX" and A for "TOP3AVG".

		RFFSu100	RFFSu10	RBF	CBFS	mRMR(2digits)
Acc.		86.92	85.78	85.73	81.67	84.65
Top 10 features	1	E2F2_v2_A	E2F2_v1_A	E2F2_v1_A	E2F2_v1_A	Irx4_A
	2	E2F3_v1_A	E2F3_v1_A	E2F2_v2_A	E2F2_v2_A	Max_M
	3	E2F2_v1_A	E2F2_v2_A	E2F3_v1_A	E2F3_v1_A	Sp4_v1_M
	4	E2F1_A	E2F7_A	E2F1_A	E2F1_A	Gata6_v1_M
	5	E2F3_v2_A	Zfp161_v2_A	E2F7_A	E2F1_M	E2F7_A
	6	E2F4_A	Zfp161_v1_A	Zfp161_v2_A	E2F3_v2_A	Sox13_v1_A
	7	E2F1_M		Zfp161_v1_A	E2F4_A	Sp100_v2_A
	8	E2F3_v1_M		Sp100_v1_A	E2F4_M	Elk4_A
	9	E2F2_v2_M		Sp4_v2_A	E2F3_v1_M	Mitf_A
	10	Zfp161_v2_A		Sp4_v1_A	E2F7_A	Sox13_v2_A

- Minimum Redundancy Maximum Relevance FS (mRMR) [27, 115]. This feature selection technique is ranking the features, considering both the relevance/importance for distinguishing between classes (which sould be as high as possible) and the redundancy between pairs of features (which sould be as low as possible). The method is based on the mutual information (MI) defined for discrete random variables, $MI(X,Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$. More specifically, let F be an initial set of n features. Feature selection can be seen as finding a subset $S \subset F$ with k features, for which the mutual information between the class C and the selected features, $MI(C,S)$, is maximal. The mRMR algorithm uses an incremental approach to optimize the condition:

$$\max_{f_i\in F\setminus S}\left\{MI(C,f_i) - \frac{1}{|S|}\sum_{f_s\in S} MI(f_s,f_i)\right\} \tag{6.1}$$

  Because this method requires dicrete features, we first performed a quantization of the features, just for the FS step, by retaining only the first two significant digits of all features' values.

We applied these three FS algorithms on the set of under 100 features selected by RF-FS, and compared the results with this initial set and with the reduced set obtained by RFFSu10. Table 6.1 presents the results in the previously discused case of c-Myc

vs. Mad (HeLa S3 cell line). We note that both RBF and CBFS algorithms fail at eliminating the biological redundancy presented in the features, while mRMR selects 9 different TFs on the first 10 positions. In the other cases studied, we observed also that RBF leads sometime to a very small number of selected features (1, 2, 3, or 4) and most of the time the accuracies are lower than those for RF-FS. For CBFS we concluded that even if the classification accuracies are comparable to those obtained for RF-FS sets, the $CScore$ is not suitable for our purpose as it has the opposite effect: related features have closely-related scores and thus the features are more clustered together by their similarity.

Thus we decided to apply mRMR and we investigated some variants of the method as well as some discretization approaches.

In the mRMR algorithm, the first term of Eq. (6.1) represents the relevance of feature f_i to the class C, while the second term measures the redundancy of feature f_i relative to all the already selected features S. The analysis of these two terms reveals that they are not comparable [36]. The first term may become significantly smaller than the second term, which may cause the selection of irrelevant but non-redundant features rather than of the relevant ones. To solve this imbalance, Estévez et al. [36] propose the use of a normalized mutual information for the second term. This normalization is conducted by the minimum entropy of both features:

$$NI(f_s, f_i) = \frac{MI(f_s, f_i)}{min\{H(f_s), H(f_i)\}} \tag{6.2}$$

The proposed algorithm is called normalized mutual information feature selection (NMIFS) and optimizes the following condition:

$$\max_{f_i \in F \setminus S} \left\{ MI(C, f_i) - \frac{1}{|S|} \sum_{f_s \in S} NI(f_s, f_i) \right\} \tag{6.3}$$

We tested both feature selection algorithms (mRMR and NMIFS) in our setting and we decided to use only the normalized version.

6.2.4.3 Discretization approaches

One challenge of using the mRMR or NMIFS algorithms in our setting is given by the fact that they both need discrete values for the features. We solved this problem by performing a temporary quantization for our features only for this feature selection component. Because these methods use mutual information (MI), and this measure is biased towards features with larger sets of possible values, quantization is not trivial.

TABLE 6.2: Top 10 features selected by mRMR with tree simple discretization techniques, sorted alphabetically, in the case of c-Myc vs. Mad (HeLa S3 cell line). Data format is similar to Table 6.1. i digit(s) – discretization that considers only the first i significant decimal(s).

		1 digit	2 digits	3 digits
Acc.		85.4	84.64	82.91
Top 10 features	1	E2F3_v1_M	E2F7_A	E2F4_A
	2	Elk4_A	Elk4_A	E2F4_M
	3	Gata3_v2_A	Gata6_v1_M	E2F7_M
	4	Gmeb1_v1_A	Irx4_A	Gata6_v1_M
	5	IRC900814_v1_A	Max_M	Max_A
	6	Irx6_A	Mitf_A	Max_M
	7	Max_M	Sox13_v1_A	Max_v1_M
	8	Mitf_A	Sox13_v2_A	Max_v2_M
	9	Otx2_A	Sp100_v2_A	Sp4_v1_M
	10	Sp4_v2_M	Sp4_v1_M	Sp4_v2_M

In order to test the MI bias we performed the following tree simple discretizations and compared the selected features by mRMR:

- 1 digit – discretization that considers only the first significant decimal
- 2 digits – discretization that considers only the first two significant decimals
- 3 digits – discretization that considers only the first tree significant decimals

We note that these approaches were applied to PBM features, which have values in $[-0.5, 0.5]$. The results for c-Myc vs. Mad are presented in Table 6.2. It is important to observe how different are the feature sets selected by the same algorithm, but with the discrete values incorporating different numbers of decimals. This effect is due to the fact that MI favors the features with greater number of distinct discrete values. The more decimals we use, more evident is this bias and instead of gaining power, the classification accuracy decreases. Moreover, the 3 digits discretization cancels the non-redundancy of mRMR FS, and only 5 different TFs are present in the top 10 set of features, with the Max TF being represented by 4 features.

Because the use of significant decimals for discretization is unreliable due to MI bias, we designed and tested more complex discretization approaches, taking into account also the biological significance of the values:

- Lin-exp – mixture of linear and exponential discretization intervals designed only for PBM features. This method converts all continuous PBM features into 20 discrete values and is based on the signification of E-scores. All the values < 0.3 (considered unbound) have a linear discretization scale with only 4 intervals, while the values > 0.3 have an exponential scale with 16 intervals (see Table 6.3).

- I-quant – independent quantile discretization, which uses 10- or 20-quantiles, computed independently for each feature, to define the discretization intervals. This method guarantees the same number of distinct values for all features, which is recommended for MI. A downside is that it uses different scales for features, and since there are features with the maximum values around 0.4 and not around 0.5, these differences between scales may be too big.
- G-quant – global quantile discretization. This approach computes 10- or 20-quantiles globally, for all values and for all features. This is possible because all features are derived from the same type of data and have very similar ranges of values. It addresses the drawback of I-quant and ensures the same scale for all features in a particular classification case, at the minor cost of possibly having slightly different numbers of distinct values for features.
- FG-quant – feature-specific global quantile discretization. It uses the global quantile discretization approach, with the only difference that the 10- or 20-quantiles are derived separately for each type of features: MAX and TOP3AVG, respectively.

We note that for all approaches based on quantiles we tested the transformation into 10 or 20 discrete values, but we are showing here only the results corresponding to 20-quantiles. Table 6.3 contains the 20-quantiles obtained for PBM features in the case of c-Myc vs. Mad, computed by the two variants of global quantile discretization.

We compared the features selected by mRMR with these four discretization techniques that deal with the cardinality of features' values (Table 6.4). The accuracies are very simmilar and the features are non-redundant, with 10 distinct TFs in each set. Based on all the cases tested, we decided to use only the feature-specific global quantile discretization. Features selected with this method have a high overlap with all the others sets. In the case of top ten c-Myc vs. Mad features, FG-quant set has 6 features in common with Lin-exp and I-quant, and 7 with G-quant.

6.2.5 Parameter optimization

We optimize the SVM & RF parameters by performing grid searches over the parameter space. Each parameter setting is evaluated with respect to the prediction accuracy, which is computed differently for the two types of algorithms. In the case of SVM we performe 5-fold cross-validation. For RF, the accuracy of each set of parameters is estimated by subtracting the out-of-bag (OOB) error fraction from 1, its maximum value. Cross-validation is not necessary because the OOB method is an unbiased subtitute for it [14].

TABLE 6.3: Discretization intervals for tree complex discretization techniques. The first column corresponds to the general values used by Lin-exp (a mixture of linear and exponential discretization intervals), while the rest present the intervals obtained for c-Myc vs. Mad (HeLa S3 cell line) with 20 quantiles in different settings: G-quant (global quantile discretization), and FG-quant (feature-specific global quantile discretization).

Lin-exp	G-quant	FG-quant	
		MAX	TOP3AVG
-0.5	-0.055	0.08	-0.055
0	0.191	0.226	0.178
0.1	0.223	0.255	0.212
0.2	0.247	0.279	0.235
0.3	0.267	0.301	0.255
$\frac{\sqrt{10}}{10} = 0.316$	0.287	0.324	0.274
$\frac{\sqrt{11}}{10} = 0.331$	0.306	0.346	0.293
$\frac{\sqrt{12}}{10} = 0.346$	0.324	0.369	0.31
$\frac{\sqrt{13}}{10} = 0.360$	0.341	0.39	0.326
$\frac{\sqrt{14}}{10} = 0.374$	0.358	0.409	0.341
$\frac{\sqrt{15}}{10} = 0.387$	0.374	0.426	0.356
$\frac{\sqrt{16}}{10} = 0.4$	0.391	0.439	0.371
$\frac{\sqrt{17}}{10} = 0.412$	0.407	0.452	0.386
$\frac{\sqrt{18}}{10} = 0.424$	0.422	0.464	0.401
$\frac{\sqrt{19}}{10} = 0.435$	0.437	0.472	0.416
$\frac{\sqrt{20}}{10} = 0.447$	0.452	0.479	0.432
$\frac{\sqrt{21}}{10} = 0.458$	0.466	0.483	0.447
$\frac{\sqrt{22}}{10} = 0.469$	0.477	0.489	0.463
$\frac{\sqrt{23}}{10} = 0.479$	0.486	0.494	0.477
$\frac{\sqrt{24}}{10} = 0.489$	0.493	0.497	0.489
$\frac{\sqrt{25}}{10} = 0.5$	0.499	0.499	0.499

For $\mathbf{SVM_{lin}}$ we optimize C, the cost of misclassifying examples. C controls the trade-off between rigid margins and misclassification: a large value of C leads to a more accurate model, but one that may not generalize well, while a small value allows for a model that has more training errors, but is not likely to over-fit the training data. The initial values considered by **COUGER** were: {0.0001, 0.00025, 0.0005, 0.00075, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1, 2.5, 5, 7.5, 10, 25, 50, 75, 100}. In order to reduce the running time, we analyzed the results for multiple datasets and we reduced the parameter values from 25 to the following 17: {0.0001, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1, 2.5, 5, 7.5, 10, 25, 50, 75, 100}.

For $\mathbf{SVM_{rbf}}$ we optimize C and the kernel parameter γ, which in LIBSVM has the default value of $1/|F|$, where F is the set of features. We tested all the 340 pairs from the following parameter sets: $C \in$ {0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1, 2.5, 5, 7.5,

TABLE 6.4: Top 10 features selected by mRMR with four complex discretization techniques, sorted alphabetically, in the case of c-Myc vs. Mad (HeLa S3 cell line). Data format is similar to Table 6.1. Lin-exp – mixture of linear and exponential discretization intervals, I-quant – independent quantile discretization, G-quant – global quantile discretization, and FG-quant – feature-specific global quantile discretization.

		Lin-exp	I-quant	G-quant	FG-quant
	Acc.	85.13	85.19	85.19	85.92
Top 10 features	1	Arid5a_v2_A	E2F3_v2_M	E2F7_A	E2F2_v2_A
	2	E2F1_A	Elk4_A	Elk4_A	Elk4_A
	3	Elk4_A	Gata6_v1_M	Gata6_v1_M	Gata6_v1_M
	4	Gata6_v1_M	Gmeb1_v1_A	Gmeb1_v1_A	Gmeb1_v1_A
	5	Gmeb1_v1_A	IRC900814_v1_A	IRC900814_v1_A	IRC900814_v1_A
	6	Hoxa2_A	Irx4_A	Irx6_A	Irx4_A
	7	Irx4_A	Max_v1_M	Max_v1_M	Mitf_A
	8	Max_v1_A	Otx2_A	Sox13_v2_A	Sox13_v2_A
	9	Sp100_v1_A	Pou3f3_A	Sp100_v1_A	Sp100_v1_M
	10	Sp4_v2_M	Sp4_v2_M	Sp4_v1_M	Sp4_v1_A

10, 25, 50, 75, 100, 125}, and $\gamma \in$ {0.00005, 0.000075, 0.0001, 0.00025, 0.0005, 0.00075, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1, 2.5}. By analyzing multiple results, we reduced the search space to only 100 pairs of values, with specific ranges for different type of input data. In the case of PBM features, **COUGER** uses $C \in$ {1, 2.5, 5, 7.5, 10, 25, 50, 75, 100, 125}, and $\gamma \in$ {0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1, 2.5}. For PWM data, the considered values are: $C \in$ {0.1, 0.25, 0.5, 0.75, 1, 2.5, 5, 7.5, 10, 25}, and $\gamma \in$ {0.00005, 0.000075, 0.0001, 0.00025, 0.0005, 0.00075, 0.001, 0.0025, 0.005, 0.0075}.

For $\mathbf{RF_{pi}}$ we optimize *ntree*, the number of trees in the forest. We note that the performance of the RF classifiers is quite stable to changes in its parameters, but certain parameter choices can significantly increase the running time.

6.2.6 Performance estimation

The performance of our framework is estimated in a 5-fold cross-validation (CV) setting. In each of the five runs, **COUGER** performs three procedures on the training data: feature selection, grid search over the parameter space, and training of the classifier(s). Then, the test data is used for prediction and evaluation. For each set of features and each type of classifier, **COUGER** computes median values for accuracy, sensitivity, specificity, and precision, using the number of true/false positive and true/false negative. Furthermore, the classification accuracy (the fraction of true positives and true negatives in the test set) is illustrated in a graphical user-friendly format.

6.2.7 Server design

We developed an extensive web implementation of **COUGER** (`http://couger.oit.duke.edu`). It runs under Apache2 (httpd.apache.org) with mod_wsgi on Debian 7.0 "Wheezy" (www.debian.org). It was developed using the Django web framework (www.djangoproject.com), which is written in Python (www.python.org) and thus allowed direct integration with the **COUGER** source code. Input data is validated using Django forms module features. Also, part of the web functions are implemented with JavaScript and jQuery (jquery.com).

6.2.8 Additional analyses

6.2.8.1 Peak calling pipeline

Our approach compares bound regions (ChIP-seq peaks) derived from multiple experiments, possibly conducted in different laboratories. For this reason, we wanted to use a uniform processing pipeline for the peak calling step. We processed the ChIP-Seq data with the IDR (Irreproducible Discovery Rate) framework [86], because it is the current standard for reporting data generated by the ENCODE project [77]. IDR is a measure of the reproducibility of findings from replicate experiments. The IDR algorithm separates signal from noise on a pair of ranked lists of ChIP-seq peaks (or other identifications) from different replicates, creating a curve which assesses the consistency and reproducibility of these peaks. The method provides stable thresholds for a list of consistency and reproducibility levels.

With regards to the actual peak calling algorithm, we reduced the candidate list to MACS [156] and SPP [69]. The selection was based on their wide usage as well as their reviewed performance. Both MACS and SPP have a high number of identified peaks and good sensitivity, but most importantly, they have a very good positional accuracy (estimated as the distance between the summit and the closest high confidence occurrence of a motif) [150].

For MACS we used two variants of version 2.0.10: one with the default estimation of the shift size, and another variant with the size of the shift previously estimated by SPP. Both variants use the parameters needed for relaxed thresholds regarding peak confidence (`-p 1e-3 --to-large`). In the case of the SPP peak caller, we used the version 1.10 with the parameters recommended by IDR usage (`-npeak=300000 -savr -savp -rf`).

We compared the peaks obtained using these three algorithms (with IDR), and two other sets of peaks available from ENCODE [35]: the peaks reported in the ENCODE database by the laboratory that performed the experiment, and the peaks reported by the ENCODE Analysis Working Group (AWG) on March 2012 Freeze that are based on IDR with SPP.

We analyzed the "quality" of a peak caller by scanning the resulted peaks for 2 consecutive 8-mers that have the PBM E-score greater than 0.4, and then plotting the percent of peaks that had at least one hit. This measure is similar to a motif scan, but uses PBM data instead of PWM data, and is highly informative for our particular setting. Moreover, because we use a fixed region centered at the summit of the ChIP-seq peaks, we considered not only the full length of the peaks in this analysis, but also the 200 bp and 300 bp region centered at the summit.

Fig. 6.3 shows the results of this analysis for c-Myc transcription factor. We can see that a large fraction of peaks contain putative TF binding sites (Fig. 6.3A), which indicates that the ChIP dataset is of high quality. Also, putative c-Myc binding sites are enriched in ChIP-seq peaks relative to open chromatin regions (i.e., DNase-seq peaks) – Fig. 6.3B. The variance between the two subplots of Fig. 6.3 is due to the length distributions of the peaks identified by different peak callers. The default ENCODE peaks are quite long in general, having up to 10,000 bp, although the majority of them have less than 1,000 bp. The SPP peak caller has the opposite behavior, and the peaks are short, usually between 80 and 350 bp, with the vast majority of peaks having around 300 bp in length. MACS2 outputs more balanced peaks, in terms of their lengths, varying between 100 and 2000 bp in length, with a large of number of peaks having between 200 and 450 bp. Therefore the analysis that considers fixed length region of peaks is more reliable, while the analysis that considers full peaks reflects mostly their length.

We analyzed the results obtained in this setting for several TFs, and we concluded that the best option for peak calling is to use MACS, but to estimate the shift size with SPP.

6.2.8.2 Using IDR to select reproducible sequences for TFs and replicates

We applied the IDR pipeline for all paralogous TFs that we considered. More precisely, for each factor, we performed peak calling (with MACS) and ran IDR analysis on several pairs of ChIP-seq mapped reads sets:

- the actual pair of replicates, obtaining the number of peaks consistent between true replicates (N_t).
- the pair of pooled pseudo-replicates, obtaining the number of peaks consistent between pooled pseudo-replicates (N_p). We generated these pooled pseudo-replicates

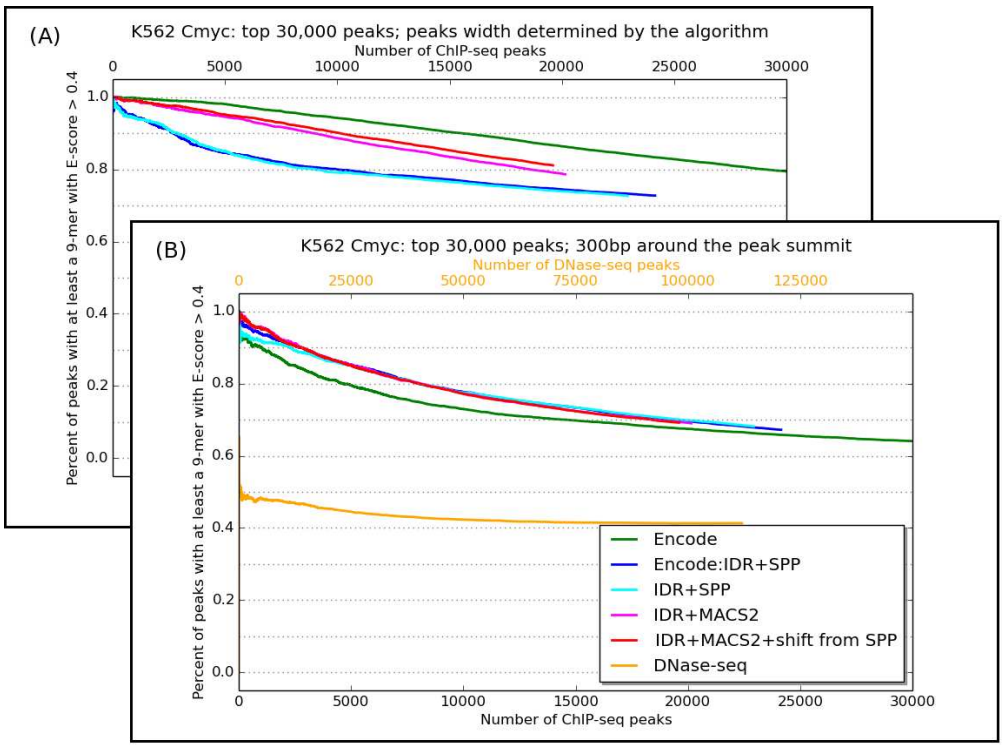


FIGURE 6.3: Peak caller comparison for c-Myc dataset: the percent of peaks with at least one high confidence binding site, for (A) peak width determined by the algorithm, and (B) 300bp around the peak summit.

by first pooling the mapped reads from all replicates and then randomly splitting this set into two equal sets of reads.

- the two pairs of self-pseudo-replicates, each corresponding to an actual replicate, obtaining the number of self-consistent peaks for each replicate (N_1, N_2). We obtained these self-pseudo-replicates by randomly splitting the mapped reads of the considered replicate into two equal sets of reads.

Next, we analyzed the results and filtered out the TFs with IDR scores that did not follow the restrictions recommended by ENCODE [77]:

$$0.5 < N_1/N_2 < 2 \tag{6.4}$$

$$N_p/N_t < 2 \tag{6.5}$$

Equation (6.4) guarantees that the number of self-consistent bound regions identified from one replicate is at least half and at most double than the number of self-consistent bound regions identified from the other replicate, while equation (6.5) ensures that the number of peaks consistent between replicates is at least 50% of the number of regions consistent between two pseudo-replicates generated by randomly partitioning available reads from all replicates.

For each considered TF, we used the IDR cutoff (N_t) to select only the peaks that are reproducible between replicate experiments. This set of high confidence peaks were then used as input in the comparisons between paralogous TFs.

For the comparisons between replicate experiments, we used N_1 and N_2 thresholds to determine self-consistent peaks for each replicate. Hence we took into account only the top N_1 peaks derived from the first replicate and the top N_2 peaks derived from the second replicate.

6.3 Results[1]

6.3.1 ENCODE ChIP-seq datasets

We tested **COUGER** on 20 pairs of paralogous TFs (Table 6.5) with ChIP-Seq data from ENCODE [35] in the K562 cell line, processed using a uniform pipeline. Briefly, we applied the IDR framework [86] together with the MACS [156] peak caller (version 2.0.10), for which the size of the shift was previously estimated by SPP [69]. Next, we analyzed the results and filtered out the TFs with IDR scores that did not follow the restrictions recommended by ENCODE [77].

Using the peak calling pipeline we identified high-quality ChIP-seq data (in terms of data reproducibility) for 20 pairs of paralogous TF (Table 6.5).

[1]This section was published as [107]

Table 6.5: Pairs of paralogous TF with high-quality ChIP-seq data for the K562 cell line in ENCODE. The pairs of factors are sorted by the "#seqs" column, which contains the number of sequences selected by **COUGER** for both classes (TF1- and TF2-specific). There are two pairs with the same TFs: CMYC and MAX. The pair marked with * corresponds to a ChIP-seq data set using an IgG control. The pair marked with ** correspond to a ChIP-seq data set using a standard control.

TF1	TF2	#seqs	TF1	TF2	#seqs
NFYA	NFYB	70	FOS	JUNB	1842
CJUN	JUND	90	ELF1	ETS1	1962
MAX	MXI1	382	ELF1	GABPA	1976
MAFF	MAFK	506	FOS	JUND	2032
CMYC	MXI1	538	JUNB	JUND	2194
CMYC	MAX*	586	CMYC	MAX**	2598
SP1	SP2	736	ETS1	GABPA	3516
GATA1	GATA2	802	CFOS	CJUN	5186
E2F4	E2F6	1002	CEBPB	CEBPD	7000
ATF1	ATF3	1682	BHLHE40	TAL1	7152

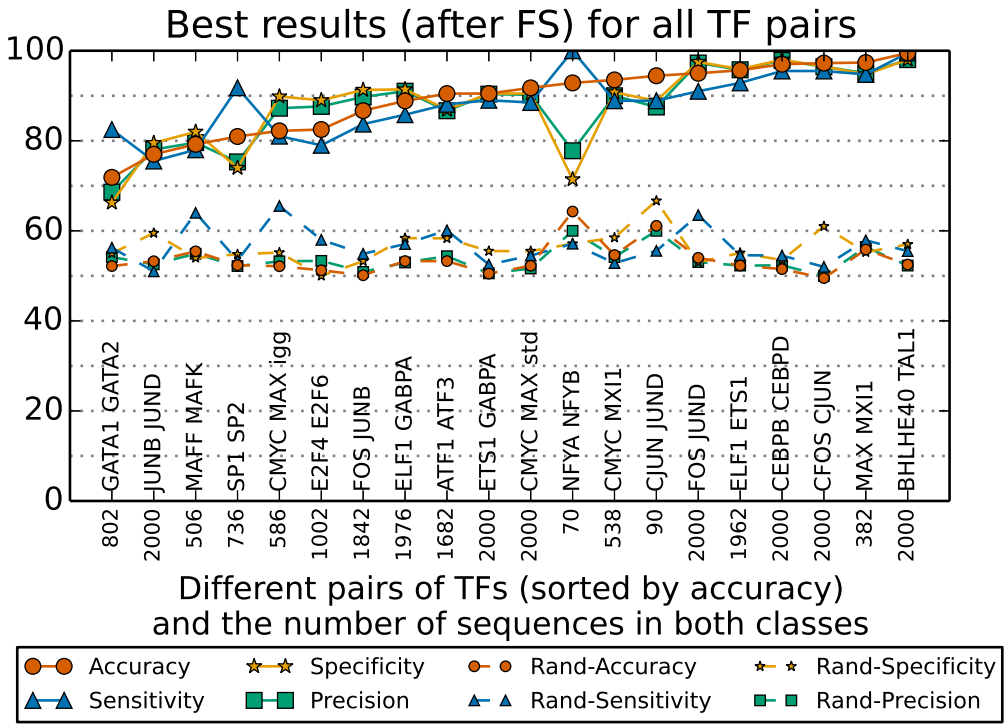


FIGURE 6.4: **COUGER** classification performance for 20 pairs of TFs, with features derived from PBM data (from UniPROBE). The values correspond to the best result from all three classifiers ($\mathbf{SVM_{lin}}$, $\mathbf{SVM_{rbf}}$ and $\mathbf{RF_{pi}}$) and all four sets of features derived by the FS procedure (under 100 and under 10 features selected by RF-FS, and first 5 and 10 features selected by NMIFS).

Having the thresholds derived from the IDR analysis, we can also redefine the TF1-specific and TF2-specific classes in a more stringent manner: the TFi-specific class consists of peaks from the set of top idr_i TFi ChIP-seq peaks that do not overlap any of the top $2 \times idr_i$ TFj-peaks, where idr_i is the determined IDR threshold for TFi, and $(i, j) \in \{(1, 2), (2, 1)\}$.

6.3.2 Classification performance

We ran **COUGER** on 20 pairs of TFs (Table 6.5), using features derived either from PBM data from UniPROBE, or the PWMs from TRANSFAC & SELEX-seq data. The classification performance for PBM features is presented in Fig. 6.4. After feature selection, the classification accuracy varies between 73.29% and 98.21% depending on the pair of paralogous TFs, with no correlation between classification performance and the number of sequences in the training set.**COUGER** performed well with both PBM-derived and PWM-derived features (Fig. 6.5, accuracy, precision, sensitivity and specificity) and returned similar sets of putative co-factors, which is not surprising given the high overlap between the TFs represented in the two data sets.

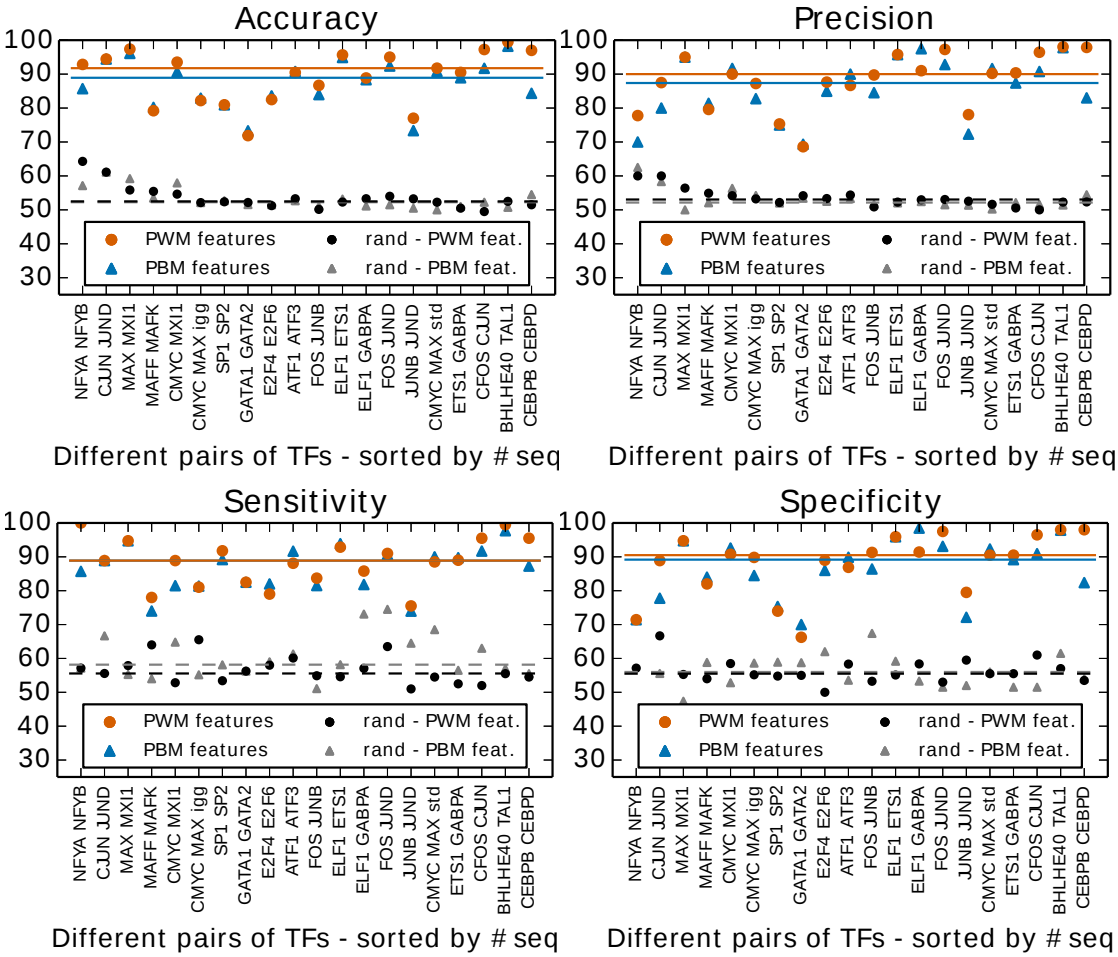


FIGURE 6.5: **COUGER** classification performance (accuracy and precision, sensitivity and specificity) for 20 pairs of TFs, with PBM and PWM features. The values correspond to the best result from all three classifiers ($\mathbf{SVM_{lin}}$, $\mathbf{SVM_{rbf}}$ and $\mathbf{RF_{pi}}$) and all four sets of features derived by the FS procedure. The horizontal lines represent the median value over all pairs of TFs, for PBM or PWM features.

To determine whether the accuracy on the tested TF pairs is significant, we randomized the classes for several pairs of TFs and ran **COUGER** on the randomized sets. As expected, the classification accuracy for randomized data varied between 44.44% and 53.70%, with a median of 49.15%, which demonstrates that the high accuracies on real TF binding data are not due to chance.

6.3.3 Identification of putative co-factors

The **COUGER** output page is highly interactive and allows users to visualize details of the classification results, as well as details regarding the co-factor features that enabled a successful classification. This is in contrast to many classification algorithms, which provide an accuracy measure but no indication of what features drive the classification.

Through its user-friendly and interactive design, **COUGER** makes it easy for users to understand and explore the contribution of individual features (see Fig. 6.6).

For several (TF1,TF2) pairs we have found evidence in the literature supporting our hypothesis that factors identified by **COUGER** interact with TF1 or TF2 and thus may contribute to their *in vivo* DNA binding differences. For example, in the case of c-Fos (henceforth referred to as Fos) and JunD (Fig. 6.6), we found support for several putative co-factors. Both Fos and JunD are basic leucine zipper (bZip) proteins from the AP-1 subfamily. Fos binds DNA as a heterodimer with c-Jun and other members of the AP-1 subfamily, while JunD can homodimerize and, interestingly, it can interact with TFs from the ATF/CREB subfamily, another branch of the bZip family. As shown in Figure 6.6, **COUGER** found that features reflecting the specificity of the ATF/CREB subfamily (such as "Atf1_3026_contig8mers_v1_MAX") are strongly associated with JunD-unique sequences, consistent with a direct interaction between JunD and ATF/CREB factors. GATA factors, also found to be important for distinguishing Fos- from JunD-unique targets, have been previously reported to interact with proteins from the AP-1 subfamily [56, 64]; the exact identity of the GATA co-factor that might interact specifically with JunD remains to be determined. TF Sfpi1, associated with JunD-unique targets, is known to interact directly with JunD [10]. TF Meis1 was also found to be associated with JunD-unique sequences. Meis1 is not known to interact directly with JunD. However, it has very similar sequence preferences to Tgif2, a factor that has been shown to interact specifically with JunD [99]. In the Fos-unique sequences, the most important features reflect the general specificity of Fos:Jun complexes, which might indicate that at those targets Fos binds together with AP-1 proteins other than JunD.

In a comparison of c-Fos versus c-Jun unique targets, we found E2F factors associated with c-Fos targets, consistent with the previously reported roles of c-Fos and E2Fs in the same signaling cascade that links Ras activity to cyclin A transcription [140]. In the same comparison, TF Mitf was associated with c-Jun, consistent with their direct interaction and synergistic effects on gene regulation [72, 130].

In a comparison of Atf1 vs. Atf3, Myc/Max/Mad TFs were found enriched in Atf3-unique targets; a potential interaction between Atf3 and Max has been reported previously in the ENCODE project and is under further investigation.

Among the tested TF pairs, we sometimes see TF1 and/or TF2 among the factors most relevant for the classification. This could indicate that one TF is present at a higher concentration, binds with higher affinity overall, or has higher quality data. Importantly, **COUGER** allows the user to remove these TFs from the set of putative co-factors (using an option in the input page) and re-run the classification framework. We note

that even after eliminating TF1 and TF2 from the set of features, the classification accuracy remains very high. For example, our results for TFs c-Myc versus Mxi1, before and after removing features reflecting the specificity of these two factors, show that the predictions accuracy, as well as the selected putative co-factors (namely, Rfx proteins) remain the same.

6.3.4 Classification between replicate experiments

Our peak calling preprocessing pipeline allowed us to determine high confidence peaks for each TFs (which allowed us to compare paralogous TFs), as well as self-consistent peaks for each replicate (which allowed us to compare replicate experiments performed for the same TF). Therefore, as a control, we also ran **COUGER** on the 31 pairs of replicate experiments. The classification accuracy for pairs of replicates ranged between 52.3% and 94.2%, with a median of 76.9%.This was far from the expected random classification result, which was obtained for the randomized classes of pairs of paralogous TFs. Randomizing the data for the replicate pairs results in an expected accuracy level (47.05%-53.84%, with a median of 50%).

By investigating further the behavior for pairs of replicates, we found that the nucleotide frequencies, in particular the GC content, plays a major role in distinguishing between sequences unique to only one replicate. Analyzing the difference in GC% for all 20 pairs of paralogous TFs and 31 pairs of TF replicates, we found that this difference correlates very well with the classification accuracy in the case of replicate experiments (Pearson correlation coefficient 0.706), but not in the case of TF pairs (Pearson correlation coefficient 0.217).The high classification accuracy for replicate data sets could be due experimental bias: recent studies have found strong biases in ChIP-seq data for regions close to the TSS of highly expressed genes [113, 142], which are oftentimes enriched in CG dinucleotides. (Indeed, many of the peaks unique to only one of the ChIP-seq replicates are close to TSSs.) Given that control experiments and replicate experiments reported in ENCODE were not performed at the same time, it is currently not possible to correct for this bias in the ChIP-seq data. We note, however, that for the TF pairs in our analysis, even when the TFs shows relatively large differences in GC-content, the identified putative co-factors were not TFs that bound CG-repeats or high GC-content sites, which suggests that the potential bias driving the classification between replicates is not influencing **COUGER**'s ability to identify co-factors.

6.4 Discussion

Identifying the molecular mechanisms that allow paralogous TFs to bind different sets of *in vivo* targets is essential for understanding eukaryotic transcription. Due to advances in high-throughput technologies for measuring TF-DNA binding both *in vivo* and *in vitro* (such as ChIP-seq and PBM), it is now possible to quantify the contributions of both intrinsic TF-DNA binding specificity and interacting co-factors to differential *in vivo* DNA binding by closely related TFs. Analyzing specific pairs of paralogous TFs, in most of the cases we identified a large number of genomic targets bound uniquely by only one of the two TFs. Considering that interactions with putative co-factors are a likely mechanism used by these proteins to select their specific genomic targets, we developed a novel framework to distinguish between genomic regions bound *in vivo* by TFs with similar binding preferences. The goal of our tool, **COUGER**, is to generate hypotheses regarding potential co-factors that provide specificity to paralogous TFs. These putative co-factors are selected from the set of TFs with known binding specificities provided as features. **COUGER** can use either PBM scores or PWM motifs for extensive sets of proteins, both types of data reflecting protein-DNA interactions.

One limitation of our proposed method is the 2-class setting for classification. Extending the approach to 3 or more classes would expand its applicability to several scenarios:

1. comparing a pair of paralogous TFs, but distinguishing between 3 classes of DNA sequences: TF1 specific sequences, TF2 specific sequences and sequences bound by both TF1 and TF2;
2. comparing more that two paralogous TFs and classifying between TF-specific genomic regions;
3. comparing more that two paralogous TFs and distinguishing between DNA sequences bound only by pairs of TFs.

Another important observation is that additional data is necessary to test whether the predicted co-factors contribute to the regulatory specificity of the paralogous TFs of interest. Given that these identified co-factors are derived from available binding motifs, several aspects should be tested:

1. whether the putative co-factors (or factors with very similar specificities) are expressed in the cell type of interest;
2. whether the identified putative co-factors are bound *in vivo* to TF1- or TF2-unique sequences; and
3. whether the putative co-factors interact physically with TF1 and TF2.

Similar classification approaches have been previously used to distinguish, for example, between ChIP-seq regions in the neighborhood of genes upregulated versus downregulated under specific conditions. Chen and Zhou [18], for example, use Naïve Bayes to identify co-factors that can distinguish between the regulatory regions of genes upregulated versus downregulated in mouse ES cells. However, they use only PWM-based features, and select the features one by one using a Wilcoxon rank-sum test. Instead, we use PBM data, which better reflects the DNA binding specificity of putative co-factors [48]. This is important because high-quality features are critical for achieving the best classification accuracy. Our choice of classification algorithms is also very important. When using features derived from TF-DNA binding specificity data (either PWMs or PBM data), it is very likely to obtain features that are highly correlated (e.g., features that correspond to paralogous TFs). While Naïve Bayes is not appropriate when features might be correlated, both SVM and RF classifiers can be used in this case. Furthermore, as the number of features increases (in our case, as more and more PBM data is being generated), a Naïve Bayes approach may start to overfit the training data, while SVMs and RFs are more robust. Finally, the advantage of using RF for feature selection (as opposed to Wilcoxon rank-sum test) is that RF can easily handle interactions among features, which would not be captured by a statistical test on individual features.

Algorithms for *de novo* motif discovery may also be used, in theory, for identifing co-factors. For example, the MEME-ChIP [94] or Peak-motifs [143] algorithms, which report several DNA motifs, could be applied to the sets of TF1- and TF2-specific targets. In this case, the expectation would be that the first motif found corresponds to the TF tested by ChIP (i.e., TF1 or TF2), while the others correspond to putative co-factors. Alternatively, methods that search for DNA motifs (PWMs) enriched in particular sets of sequences (e.g., SpaMo [149] or AME [100]) could also be used to identify putative co-factors. Both these approaches would only search for one co-TF at a time, and would be able to find only DNA motifs that appear in a significant fraction of the DNA sequences of interest (here, the TF1- or TF2-specific sequences). However, recent evidence from the ENCODE project shows that the co-association of TFs is highly context specific, i.e., distinct combinations of TFs bind at specific genomic locations [47]. Thus, classification approaches that search for sets of putative co-factors in TF-specific genomic targets are more likely to reveal important molecular mechanisms through which paralogous TFs achieve their regulatory specificity in the cell.

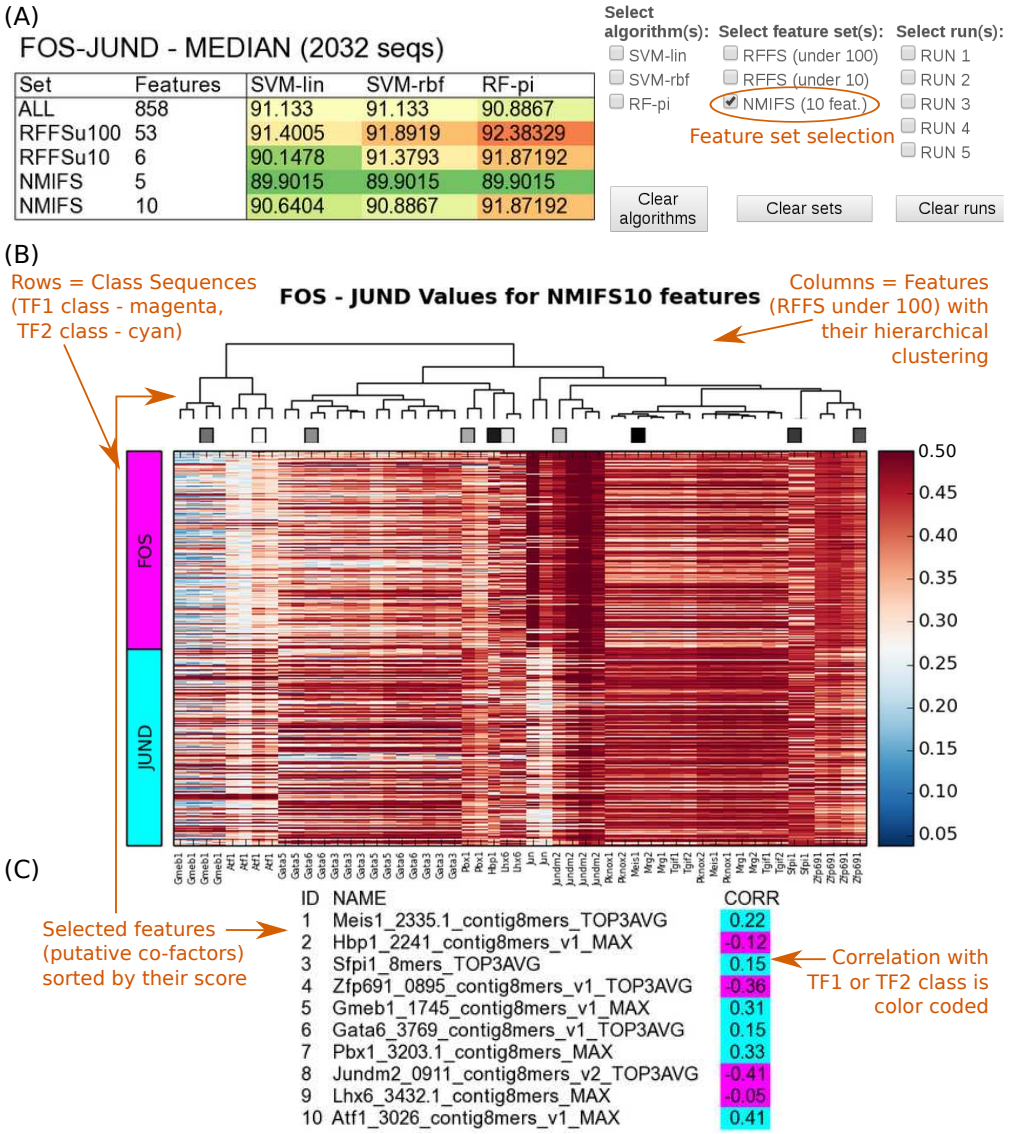


FIGURE 6.6: **COUGER** output for TFs Fos and JunD, with PBM-derived features. (A) Median classification accuracies for Fos vs. JunD (left), and options for interactive selection (right). (B) Heatmap showing the features values. Each row represents a DNA sequence in one of the two classes. Each column represents a selected feature from "RFFS under 100" (i.e., Random Forest feature selection run to select < 100 features). (C) A set of selected features (sorted by their score), together with their correlation (i.e., the Pearson correlation coefficient) with the class label. The first class, in this example Fos, is considered class "0". The second class, in this example JunD, is considered class "1". Thus, a negative correlation for a particular feature suggests that the feature is important for TF1, while a negative correlation suggests that the feature is important for TF2. The name of each feature contains the name of PBM file used to generate that feature, as well as "MAX" or "TOP3AVG", which specifies whether the feature represents the score of the best site in each sequence or an average over the top three sites, respectively.

Chapter 7

SSMART: an algorithm for *de novo* RNA motif identification

7.1 Introduction[1]

RNA-binding proteins (RBPs) are key players in RNA metabolism and function. They bind RNA molecules through cis-regulatory elements to coordinate post-transcriptional processes such as splicing, RNA transport, RNA stability and localization [68]. There are hundreds of RBPs encoded in eukaryotic genomes [9], each with specific functions, thus it is necessary that the RBPs recognize their RNA targets with high specificity. The current understanding is that this binding specificity is achieved through combinations from RNA sequence and structure properties, in variable proportions [20]. Some RBPs prefer to bind single-stranded RNA and recognize their target only by the nucleotide composition, while others prefer specific structural contexts. For most RBPs, however, only a primary sequence motif has been determined, while the structure of the binding sites is uncharacterized.

RBP-RNA interactions are experimentally assessed with high-throughput *in vitro* or *in vivo* methods. The *in vitro* approaches, like RNAcompete [120], determine the binding specificity and affinity of a specific protein to millions of short, synthetic RNAs, in the absence of other proteins or cellular factors, while the *in vivo* methods ascertain the binding sites of a certain protein in a specific cellular context. There are a number of crosslinking and immunoprecipitation (CLIP) methods [53, 76, 145] that induce permanent cross-links between RNAs and RBPs *in vivo*, after which the RBP-RNA fragments are isolated using immunoprecipitation, and the crosslinked RNA segments are sequenced.

[1] This section was published as [106]

Computational analysis of RBP-RNA interactions is vital for interpreting the experimental data and finally understanding how an RBP finds and binds to its targets. Finding sequence motifs is not trivial due to the shortness of the binding motif and the large number of input sequences that can include many false positives. Incorporating secondary structure preferences into motif models adds an extra layer of challenges due to the noisiness of RNA structure prediction and the need of a reliable model for sequence-structure motifs that is also easy to interpret. Available tools that model and predict both sequence and structure motifs for RBs include *RNAcontext* [66], *GraphProt* [97] and *Zagros* [7]. These motif finders are design specifically for a certain type of experimental data and only *Zagros* finds *de novo* sequence-structure motifs, while *RNAcontext* and *GraphProt* work in classification or regression settings. Furthermore, the structure predictions of all tools were not objectively evaluated.

In this chapter, we present **SSMART** (sequence-structure motif analysis tool for RNA-binding proteins), an RNA motif finder that extends cERMIT [46] – a sequence-based motif finder used primarily to determine DNA binding preferences from high throughput data such as CHIP-seq. Our tool uses this approach to identify binding motifs, simultaneously modeling the primary sequence and the secondary structure of the RNA. The sequence-structure motifs are represented as consensus strings over a degenerate alphabet, extending the IUPAC codes for nucleotides to also reflect secondary structure preferences. More specifically, **SSMART** takes as input putative RBP binding sites, described in terms of sequence and structure, and scores that reflect the binding strength, and searches for optimal sequence-structure motifs of flexible lengths. The secondary structure is obtained in a prior step, by sampling suboptimal structures around binding sites and identifying local dominant combinations of base pairs [128]. The motif candidates are evaluated with an objective function that integrates the individual RNA targets binding evidence into a combined score. There are two available scoring functions: one based on random set scoring, and the other in a linear regression framework. The objective function is optimized with a greedy search strategy that starts with a set of 4-mers over the non-degenerate sequence-structure alphabet. Each of this "seed" motifs is then "evolved" iteratively until the motif score cannot be improved anymore. After all the motif seeds are evolved, **SSMART** applies a post-processing step in which the evolved motifs are clustered and merged, and corresponding PWMs are generated from high scoring candidates (Fig 7.1).

Evaluations on synthetic data showed that **SSMART** is able to recover RBP sequence and structure motifs implanted into 3'UTR-like sequences, in various proportions of structured/unstructured binding sites. We successfully used **SSMART** on high-throughput *in vivo* and *in vitro* data, showing that we not only recover the known sequence motif, but also gain insight into the structural preferences of the RBP.

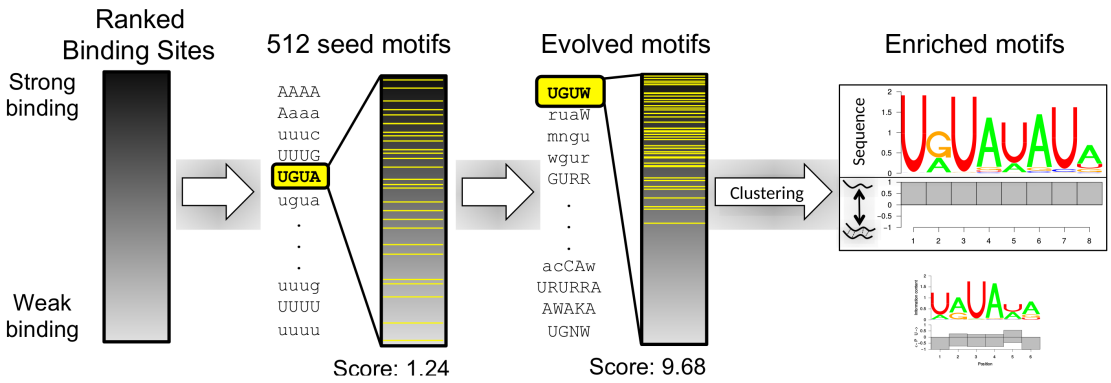


FIGURE 7.1: The **SSMART** framework. A set of 4-mer seeds are independently evolved in order to optimize the score over the ranked list of binding sites. Then the motifs are clustered and the best sequence-structure motifs are reported. We represent the two components separately: the upper part corresponds to the sequence logo, while the lower part depicts the probability to be paired (below the line) and unpaired (above the line) for each base.

7.2 Methods[2]

In this section we describe in detail the motif finding strategy implemented in **SSMART** as well as the employed evaluation procedures. We also explain the datasets that were used to test our tool, including how the synthetic ones were generated.

7.2.1 RNA secondary structure prediction

RNA molecules are flexible oligonucleotides that can adopt multiple stable structures. Their folding is influenced not only by their composition and the local environment, but also by other molecules, so it is difficult to obtain the exact secondary structure that an RNA has during an interaction with a protein *in vivo*. The available structure prediction tools are based either on free energy minimization [12, 159], or on ensembles of secondary structures [28, 128]. The more recent algorithms focus on local conformations and can take into account multiple suboptimal structures.

In this section we considered two folding algorithms: RNAplfold [12] and RNAprofiling [128], and we compared their results on binding sites for two proteins with different structural preferences: PUM2, an RBP that prefers ss-RNA; and Staufen1, known to bind ds-RNA. We predicted RNA secondary structures for PUM2 and Staufen1 binding sites derived from *in vivo* PAR-CLIP and RIPiT data, respectively [53, 124]. We extended the core regions with maximum 25, 50, 75 or 150 nucleotides on each side, using either genomic coordinates in the case of intronic sequences, or the highest expressed isoform that contains the peak for the exonic ones. First, we compared RNAplfold predictions

[2]This section was largely published as [106]

in multiple parameter settings, and then we compared the RNA secondary structure predictions of RNAplfold vs. RNAprofiling.

7.2.1.1 RNAplfold predictions depend on parameters

RNAplfold is a tool from ViennaRNA package that predicts RNA single-strandedness using free energy minimization and locally stable secondary structures. It associates the best structure to each sliding window over the stretch of RNA of interest, and then outputs the average base pair probabilities. RNAplfold has two important parameters: the size of the window (W) and the maximum base pair span (L). Multiple applications of RNAplfold use the values $(W, L) = (80, 40)$ (see [66, 83, 88, 95]), while [78] recommend that $W = L + 50$ in order to have each base present in at least 51 windows.

We selected a wide range of parameters: $L \in [30, 150]$ and $W \in [L, L + 100]$, resulting in 143 (W, L) pairs, and we applied RNAplfold on 3974 PUM2 and 4666 Staufen1 peaks. In the folding step, we extended the peaks with maximum 150 bp on each side, after which we discarded the flanking regions and we analyzed only the initial binding sites with the associated RNA structures. Fig. 7.2 shows the percent of bases that were predicted to correspond to ds-RNA (bases that have the predicted unpaired probability < 0.5). RNAplfold predicts on average 13% more paired bases for Staufen1 binding sites than for PUM2 binding sites, a result consistent with the reported binding preferences for the two RBPs. We note that in both sets the results show a clear trend of more base pairs as the parameter values increase. This is also in agreement with the reported behaviour of multiple structure prediction tools, namely that the prediction accuracy for individual base-pairs decreases with respect to span length and/or window length [30, 73, 78]. Given this correlation between the parameter size and the percent of paired bases for both sets, it is unclear what the "optimal" parameter values might be. Looking at just two widely used parameter values, $(W, L) \in \{(80, 40), (150, 100)\}$, we observe an unsettling difference of 11.1% for PUM2 and 9.3% for Staufen1 in the number of paired RNA bases.

7.2.1.2 Comparative analysis

Given the performance of RNAplfold with different parameter settings, we looked for another tool for RNA structure prediction, and we selected RNAprofiling, which has a different approach to RNA folding. RNAprofiling is an ensemble-based method that balance abstraction and specificity by identifying local dominant combinations of base pairs. It uses a statistical sample of 1000 RNA secondary structures from the Boltzmann ensemble of possible RNA secondary structures associated with a given RNA sequence.

A PUM2 percent of paired bases

W \ L	30	40	50	60	70	80	90	100	110	120	130	140	150
L	0.187	0.279	0.337	0.376	0.408	0.43	0.448	0.462	0.474	0.483	0.49	0.496	0.501
L+10	0.243	0.313	0.359	0.394	0.418	0.438	0.453	0.466	0.476	0.483	0.491	0.496	0.506
L+20	0.276	0.334	0.376	0.405	0.426	0.444	0.458	0.47	0.478	0.485	0.491	0.502	0.513
L+30	0.304	0.358	0.393	0.417	0.437	0.453	0.465	0.476	0.483	0.489	0.499	0.512	0.525
L+40	0.321	0.374	0.405	0.428	0.446	0.46	0.47	0.481	0.488	0.496	0.509	0.523	0.532
L+50	0.331	0.384	0.415	0.436	0.452	0.465	0.475	0.485	0.494	0.507	0.521	0.53	0.537
L+60	0.338	0.391	0.422	0.442	0.458	0.47	0.48	0.492	0.505	0.518	0.528	0.536	0.54
L+70	0.344	0.397	0.427	0.447	0.462	0.474	0.487	0.502	0.516	0.526	0.533	0.538	0.542
L+80	0.348	0.401	0.431	0.45	0.465	0.481	0.497	0.514	0.524	0.531	0.537	0.541	0.544
L+90	0.352	0.405	0.434	0.453	0.471	0.492	0.509	0.522	0.529	0.534	0.539	0.543	0.546
L+100	0.355	0.408	0.436	0.459	0.482	0.503	0.517	0.527	0.533	0.537	0.541	0.544	0.548

B Staufen1 percent of paired bases

W \ L	30	40	50	60	70	80	90	100	110	120	130	140	150
L	0.372	0.451	0.498	0.53	0.553	0.571	0.582	0.593	0.603	0.609	0.618	0.623	0.629
L+10	0.412	0.473	0.511	0.539	0.56	0.573	0.585	0.596	0.604	0.612	0.619	0.624	0.632
L+20	0.443	0.491	0.525	0.549	0.566	0.579	0.59	0.599	0.608	0.615	0.621	0.629	0.635
L+30	0.463	0.505	0.536	0.557	0.573	0.585	0.595	0.604	0.611	0.617	0.625	0.632	0.639
L+40	0.474	0.516	0.544	0.563	0.578	0.59	0.599	0.606	0.613	0.621	0.628	0.637	0.642
L+50	0.481	0.522	0.55	0.568	0.582	0.593	0.601	0.609	0.617	0.626	0.633	0.639	0.645
L+60	0.487	0.526	0.554	0.571	0.586	0.596	0.604	0.613	0.621	0.63	0.636	0.642	0.649
L+70	0.491	0.529	0.557	0.574	0.588	0.598	0.609	0.617	0.626	0.633	0.638	0.645	0.652
L+80	0.493	0.533	0.559	0.576	0.59	0.602	0.613	0.622	0.629	0.635	0.641	0.648	0.656
L+90	0.496	0.535	0.561	0.578	0.593	0.606	0.617	0.625	0.632	0.638	0.644	0.651	0.658
L+100	0.498	0.536	0.563	0.581	0.597	0.61	0.619	0.628	0.634	0.641	0.647	0.653	0.661

FIGURE 7.2: RNAplfold predictions with different values for parameters W (window length) and L (maximum base pair span). The numbers presented in a heatmap-like manner correspond to the percent of paired nucleotides. The values in boxes correspond to three parameter settings widely used in the literature: $(W, L) \in \{(80, 40), (150, 100), (200, 150)\}$. (A) Predictions for PUM2 binding sites. (B) Predictions for Staufen1 binding sites.

The tool then focuses on the arrangement of helices at the substructure level and reports the most frequent double-stranded regions. We consider a base to be paired only if it is contained in a helix that is present in more than half of the ensemble of secondary structures. RNAprofiling has no parameters, the results being influenced only by the length of the input sequence. We tested it with three different sizes for the flanking regions of the RBP binding sites: 25, 50 and 75 and we compared the results with RNAplfold predictions for $(W, L) \in \{(80, 40), (150, 100), (200, 150)\}$.

We used the Paired/Unpaired predictions to define the following similarity metric between two foldings a, b of the same sequence of length w:

$$Sim(a, b) = \frac{1}{w} \sum_{i=1}^{w} s_i, \text{ with } s_i = \begin{cases} 1, & \text{if } a_i \neq b_i \\ 0, & \text{if } a_i = b_i \end{cases} \tag{7.1}$$

where $a = \{a_1, a_2, \ldots, a_w\}$ and $b = \{b_1, b_2, \ldots, b_w\}$. A similarity score of 0 denotes identical structures, while a score of 0.3 means that the two predicted structures disagree

in 30% of their positions. We compared the structures predicted by the two tools, each in three settings, for PUM2 and Staufen1 binding sites. The average similarity scores are depicted in Fig 7.3A. We note that the similarity between structure predictions for PUM2 (above the main diagonal) and Staufen1 (below the diagonal) has the same trend, with smaller values for comparisons between the same tool or between parameter settings with simmilar window lengths. The results that disagree the most (around 30%) correspond to the following pairs:

- RNAprofiling with shortest sequences (25 bp padding) & RNAplfold with the longest parameters ($W = 200, L = 150$), and
- RNAprofiling with longest sequences (75 bp padding) & RNAplfold with the shortest parameters ($W = 80, L = 40$).

The structural predictions that agree the most acros different tools (23-24%) correspond to simmilar folding sequences:

- short sequences: RNAprofiling with 25 bp padding & RNAplfold with $W = 80, L = 40$, and
- mid-range sequence: RNAprofiling with 50 bp padding & RNAplfold with $W = 150, L = 100$.

We also analyzed the percent of bases that were predicted to correspond to ds-RNA (Fig 7.3B). Staufen1 binding sites have an average 12% more paired bases than PUM2 sites, a trend consistent with the binding preferences shown by these RBPs. However, the secondary structure predicted by RNAplfold is correlated with the parameters used, longer values yielding more paired bases, while RNAprofiling predictions are quite stable with respect to the length of flanks.

We used RNAprofiling for all **SSMART** results reported here, but the user can compute secondary structures with any tool, if then the predicted structures are properly encoded into the input sequences. In order to predict secondary structures relevant to the experimental data, we apply the folding algorithm either on the RNA oligos in the case of RNAcompete data, or on extended peaks in the case of CLIP datasets ($\pm$50 bp). The extension is performed using either genomic coordinates in the case of intronic sequences, or the highest expressed isoform (from RNA-seq data) that contains the peak for the exonic ones. The extended RNA sequences are used only in the structure prediction step, after which only the core sequences are associated with the secondary structures in **SSMART** input files.

A

RNAprofiling vs RNAplfold - similarity

Staufen1	PUM2	RNAprofiling P25	RNAprofiling P50	RNAprofiling P75	RNAplfold W80 L40	RNAplfold W150 L100	RNAplfold W200 L150
RNAprofiling	P25	0.00	0.22	0.26	0.23	0.26	0.29
	P50	0.25	0.00	0.19	0.26	0.23	0.25
	P75	0.28	0.20	0.00	0.28	0.24	0.25
RNAplfold	W80.L40	0.24	0.28	0.31	0.00	0.20	0.25
	W150.L100	0.27	0.24	0.24	0.21	0.00	0.11
	W200.L150	0.30	0.26	0.25	0.24	0.12	0.00

B

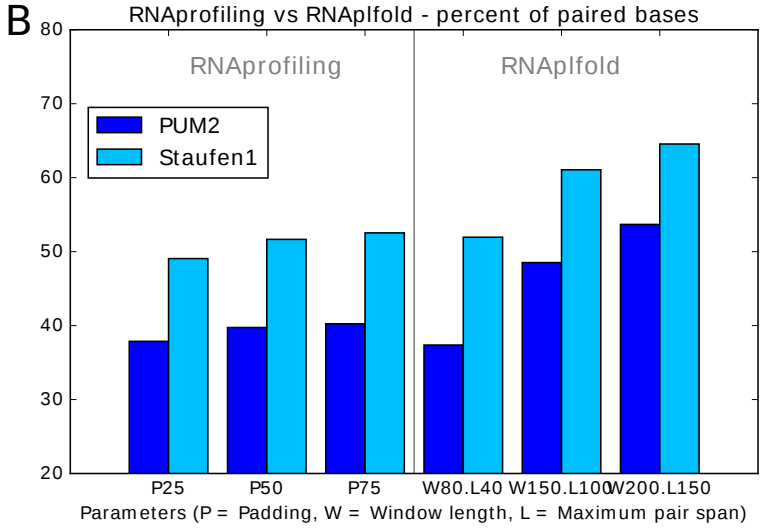


FIGURE 7.3: RNAprofiling vs RNAplfold in different parameter settings (P = Padding, W = Window length, L = Maximum pair span). (A) Average similarity between predicted structures. The values above the diagonal correspond to PUM2, while the values below the diagonal correspond to Staufen1. (B) Percent of paired bases for PUM2 and Staufen1 binding sites.

7.2.2 The sequence-structure motif identification framework

In order to simultaneously model the primary sequence and the secondary structure of the RNA, **SSMART** represents the sequence-structure motifs as consensus strings over an extended degenerate alphabet. We use the regular IUPAC codes for nucleotides to denote bases in single-stranded positions, and their lower-case counterparts to denote bases in double-stranded context.

Given a set of putative RBP binding sites with corresponding binding scores for each site, **SSMART** searches for optimal sequence-structure motifs of flexible lengths (Fig 7.1). The framework has two essential components: an objective function that scores the binding strength of a given k-mer and a search procedure that explores the motif space for high-scoring k-mers.

A **SSMART** motif of length k is a k-mer over the alphabet $A_{complete}$ = {A, C, G, T, W, K, R, Y, S, M, N, a, c, g, t, w, k, r, y, s, m, n}. We define the motif space to be all

k-mers with length between 4-10 over the $A_{complete}$ alphabet, with a limited number of degenerate positions.

7.2.2.1 Binding evidence

SSMART input consists in the set of n input sequences $s_i, i = 1, \ldots, n$ (for example, CLIP peaks or RNAcompete oligos), described with the sequence-structure 8 letters alphabet A_{basic} = {A, C, G, T, a, c, g, t}, and their corresponding binding scores $y_i, i = 1, \ldots, n$. These scores depend on the type of experiment used to derive the binding specificities of the RBP in question. In the case of CLIP experiments, we used PARalizer peaks together with cell line-specific gene expression from RNA-seq data to define the following binding scores: normalized read counts $y_i = \log_{10} \frac{\#\text{ reads}}{\text{gene expression}+0.01}$ for CLIP-seq datasets; and normalized T-to-C conversion specificity $y_i = \log_{10} \frac{\#\text{ reads with T-to-C conversions}}{(\#\text{ reads with other conversions}+1)(\text{gene expression}+0.01)}$ for PAR-CLIP datasets. For RNAcompete datasets we used the affinity scores (normalized signal intensities) to describe the binding preferences. We note that in the case of *in vivo* CLIP experiments the majority of sequences in the dataset correspond to binding events, while for *in vitro* RNAcompete data the majority of input sequences will be unbound. **SSMART** is able to handle both types of score distributions.

7.2.2.2 The objective function

There are two available approaches for evaluating the binding strength of a given k-mer in our framework: a random set score and a linear regression score.

The random set approach (RS score) was described in [46] and works well with a variety of scores that reflect the direct binding evidence, but is restricted by the assumption of independent contributions for the space of the input sequences. Given the sequences s_i and scores y_i, a motif m_j partitions the sequence space into a positive set (containing m_j) and a negative set (not containing m_j). We search over the discrete space of possible motifs for the optimal motif m^* that yields high binding scores in the positive set and low binding scores in the negative set. Given a motif m_j, we denote the number of sequences with motif occurrences with $n_j = \sum_{i=1}^{n} x_{ij}$, where the binary variable x_{ij} indicates a match of m_j in sequence s_i. We consider the enrichment score $X(m_j) = \frac{1}{n_j} \sum_{i:x_{ij}=1} y_i$ as a random variable whose randomness comes through the set of sequences containing m_j ($x_{ij} = 1$), and not through the scores y_i, and we define the random set scoring function

to be its z-score:

$$S_j^{RS} = S^{RS}(m_j) = \frac{X(m_j) - \mu}{\sigma_j}, \text{ where} \tag{7.2}$$
$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i, \sigma_j^2 = \frac{n - n_j}{n_j(n-1)}\left[\frac{1}{n}\sum_i y_i^2 - \left(\frac{1}{n}\sum_i y_i\right)^2\right]$$

This is a zero mean, unit variance test statistic on the null hypothesis that sequences containing m_j are not enriched for the motif m_j. The optimization problem is:

$$\hat{m}^{RS} = \arg\max_{m_j \in M} S_j^{RS} \tag{7.3}$$

where M is the set of putative motifs $\{m_1, \ldots, m_p\}$, and $\hat{m}^{RS}$ is the best guess at the optimal binding motif m^*.

We can extend the scoring function to any rule $R(s_i)$ that partitions the sequence space into two sets as follows: We denote by x_{iR} the truthfulness of rule R in sequence s_i. Then $n_R = \sum_{i=1}^{n} x_{iR}$, $X(R) = \frac{1}{n_R}\sum_{i:x_{iR}=1} y_i$, and the induced score is

$$S^{RS}(R) = \frac{X(R) - \mu}{\sigma_R} \tag{7.4}$$

The score from Eq. (7.2) is a particular case, with the rule $R(s_i) = (m_j \subset s_i)$.

The linear regression approach (LR score) was introduced in [21] and is more computationally demanding but can account for some, potentially relevant, confounder information, like di-nucleotide frequencies or sequence length. In order to describe the regression framework, we will rewrite the random set score from Eq. (7.2). We make the following notations: $\mu = \frac{1}{n}\sum_{i=1}^{n} y_i$, $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu)^2$, $y_i^* = y_i - \mu$, $A_j = \frac{n-n_j}{n-1}$, $e_j = \frac{1}{n_j}\sum_{i:x_{ij}=1} y_i^*$, $\hat{\sigma_j}^2 = \frac{\sigma^2}{n_j}$. Then we have $S_j^{RS} = A_j \times \frac{e_j}{\hat{\sigma_j}}$ with the top predicted motif $\hat{m}^{RS}$ described by Eq. (7.3).

Now we can define a linear regression model for the binding interactions, that closely resembles the random set scoring strategy described above. Let the regression coefficient for motif m_j be denoted β_j, then a simple linear model for binding is:

$$y_i^* = x_{ij}\beta_j + \epsilon_i, \text{with } \epsilon_i \sim N(0, \sigma^2) \tag{7.5}$$

We estimate the regression coefficient with classical ordinary least squares (OLS): $\beta_j^{OLS} = \frac{1}{n_j}\sum_{i:x_{ij}=1} y_i^*$ and we define the motif enrichment score as follows:

$$S_j^{LR} = \frac{\beta_j^{OLS}}{\hat{\sigma_j}} = \frac{1}{A_j} \times S_j^{RS} \tag{7.6}$$

The top motif is in this case: $\hat{m}^{LR} = \arg\max_{m_j \in M} S_j^{LR}$. Note that if the size of the motif target set is small relative to the number of all input sequences ($n_j << n$), $A_j \approx 1$ and $S_j^{LR} \approx S_j^{RS}$.

This framework can be extended to account for features of the input sequences that may be unrelated to motif binding. One such potential confounder is the sequence length, and other important features can be derived from the nucleotide content, like the di-nucleotide counts. If we consider q additional confounders, the regression covariates for motif m_j are defined by the following matrix: $Z_j = (x_j, c_1, c_2, \ldots, c_q)$, where x_j represents the column vector of motif matches x_{ij}, and $c_k \in \mathbb{R}^n, k \in \{1, \ldots, q\}$ are the confounders. We denote the corresponding regression coefficients with $\beta_{0j}, \beta_{1j}, \ldots \beta_{qj}$, with $\beta_{kj} \in \mathbb{R}, k \in \{0, \ldots, q\}$. Then the model can be expressed with the matrix notation as:

$$Y^* = Z_j \beta_j + \epsilon, \text{where } \epsilon \sim N(0, \sigma^2 I_{n \times n}) \tag{7.7}$$

In the typical case there are a large number of input sequences and therefore a large number of sample points to use in the estimation process. Also, we need to score a large number of motif candidates, therefore we need a computationally efficient estimator for the regression coefficients, with good statistical properties. We use the simple OLS estimator $\beta_j^{OLS} = (Z_j^T Z_j)^{-1} Z_j^T Y^*$, that provide fast solutions for small to moderate q values. The corresponding motif scoring function is a straightforward generalization of the univariate regression case:

$$S_j^{LR} = \frac{\beta_j^{OLS}}{\hat{\sigma_j}}, \text{with } \hat{\sigma_j}^2 = (\hat{\Sigma}_{\hat{\beta_j}})_{11} \tag{7.8}$$

where $(\hat{\Sigma}_{\hat{\beta_j}})_{11}$ represents the first diagonal element in the covariance matrix for the parameter estimate β_j^{OLS}.

7.2.2.3 The search strategy

We need to search the sequence-structure motif that optimizes one of the objective functions defined before (Eq. 7.2 or Eq. 7.8). An exhaustive search over the space of all potential motifs is not computationally feasible, thus we employ a custom greedy search strategy that considers a large set of seed motifs that are independently updated. These seed points are motifs of lenght 4 with the same structure and all possible sequence composition (512 4-mers over the alphabets $\{A, C, G, T\}$ and $\{a, c, g, t\}$). We note that we obtain consistently similar results with this (reduced) set as with the whole set of 4096 possible 4-mers over the A_{basic} alphabet.

Given a motif m, a set of candidate motifs is constructed by applying small variations to m: in length, sequence, or structure. The k-mer m is extended to 16 new $(k+1)$-mers, by independently adding one letter from A_{basic} at one of its end. If $k > 4$, the length of the motif is reduced and 2 new $(k-1)$-mers are considered. Then a large set of new k-mers are obtained by changing one letter at a time in terms of structural change or increasing/decreasing sequence degeneracy. Briefly, for each position j in m, the following rules are applied:

- if $m[j] \in \{A, C, G, T, a, c, g, t\}$ then four new motif candidates are constructed by replacing $m[j]$, in turns, with its structural complement and the three letters that encode for it and one other nucleotide, keeping the original structure; for example A will become a, M, R, W, and g will become G, k, r, s;
- if $m[j] \in \{W, K, R, Y, S, M, w, k, r, y, s, m\}$ then three new k-mers are obtained with $m[j]$ set to either its structural complement, or to one of the nucleotides it encodes, in the same structural context; for example R will become r, A, G;
- if $m[j] \in \{W, K, R, Y, S, M, w, k, r, y, s, m\}$ and $k \notin \{1, k\}$ then a fourth candidate is obtained by setting $m[j] = N$ or $m[j] = n$, depending on the case;
- if $m[j] \in \{N, n\}$ then 10 new motifs are considered by changing $m[j]$ to one letter from either $\{A, C, G, T, W, K, R, Y, S, M\}$, or $\{a, c, g, t, w, k, r, y, s, m\}$.

For example $(4k + 18)$ candidate motifs are generated for a k-mer with no sequence degeneracies, or for a k-mer with double degeneracy in two middle positions. In the case of a k-mer with double degeneracy at its ends, $(4k + 16)$ new motifs are considered.

Each seed motif m_i starts an independent search for the best motif. At one iteration, all update rules are applied and each new motif candidate is scored. The motif candidate with the highest motif score is used in the next iteration. The procedure is repeated until the motif score cannot be improved, in which case the last motif is reported. The result of the search is a set of evolved motifs $\hat{M} = \{\hat{m}_1, \ldots, \hat{m}_p\}$ and their corresponding scores $\hat{S} = \{\hat{S}_1, \ldots, \hat{S}_p\}$. For each motif $\hat{m}_i$ a PWM is derived from the empirical counts of occurrences of each of the exact instantiations in the top 50% input sequences.

7.2.2.4 Post-processing procedure

The complete set of evolved motifs will have many similar motifs that vary by a few letters or have different lengths and/or overlap. **SSMART** applies a post-processing procedure in order to cluster multiple evolved motifs together and to rank these merged motifs. We use the metric introduced by [55] to define a similarity measure as follows. For two motifs a, b of equal length w, the Harbison distance is $D(a, b) =$

$\frac{1}{\sqrt{2}w}\sum_{i=1}^{w}\sqrt{\sum_{L\in A}(a_{i,L}-b_{i,L})^2}$, where A is the alphabet and $a_{i,L}$, $b_{i,L}$ are the probabilities of observing base L at position i of motifs a and b, respectively. As in [46], we define the following similarity score:

$$sim(a,b)=\max_{a',b'}[1-D(a',b')] \tag{7.9}$$

where a', b' correspond to all possible overlaps between motifs a, b induced by shifts such that the minimum overlap length is 3.

Given the set of redundant output motifs $\hat{M}=\{\hat{m}_1,\ldots,\hat{m}_p\}$, we obtain a set of ordered motif clusters $\{C_i\}$ with the following clustering procedure:

1. Initialize the cluster count: $q=1$;
2. Find the top motif in $\hat{M}$: $m^*=\arg\max_{\hat{m}_j\in\hat{M}}\hat{S}_j$;
3. Select all motifs $m_j\in\hat{M}\setminus\{m^*\}$ with $sim(m^*,m_j)\geq 0.75$ and compute scores for the union rule $R_{*|j}(s_i)=(m^*\subset s_i)|(m_j\subset s_i)$ and the intersection rule $R_{*\&j}(s_i)=(m^*\subset s_i)\&(m_j\subset s_i)$;
4. Add m^* and all similar motifs m_j that have $S(R_{*|j})\geq 0.95\cdot S(m^*)$ or $S(R_{*\&j})\geq 0.95\cdot S(m^*)$ to C_q;
5. Remove the set C_q from $\hat{M}$;
6. Update cluster count: $q=q+1$;
7. Repeat steps 2 to 6 until $\hat{M}$ is empty.

For each motif cluster C_i we compute an aggregate PWM by averaging the PWMs of each cluster member weighted by its motif score.

For a better understanding of the clustering step, **SSMART** plots all unique evolved motifs and the similarity between them in two ways: as a heatmap and as a network graph (see Fig. 7.4). The similarity is computed with the metric defined for the post-processing step. The heatmap plot depicts all pair-wise similarities between the evolved k-mers, together with their hierarchical clustering. The network graph provides a different perspective of the same data, filtering out the low similarities. In both the heatmap and the network graph the motifs are colored according to the motif cluster they belong to after the custom clustering procedure.

Fig. 7.4 contains the visualization of k-mers similarity for two libraies corresponding to PUM2 and FUS proteins. While for PUM2 the evolved motifs are more homogeneous, the ones for FUS apear to be more disperse. Depending of the data, there can be small motif clusters (like Motif2 of PUM2 with 4 k-mers or Motif3 of FUS with 5 k-mers), or clusters that incorporate many evolved motifs (like the top PUM2 motif that contains almost half of the k-mers).

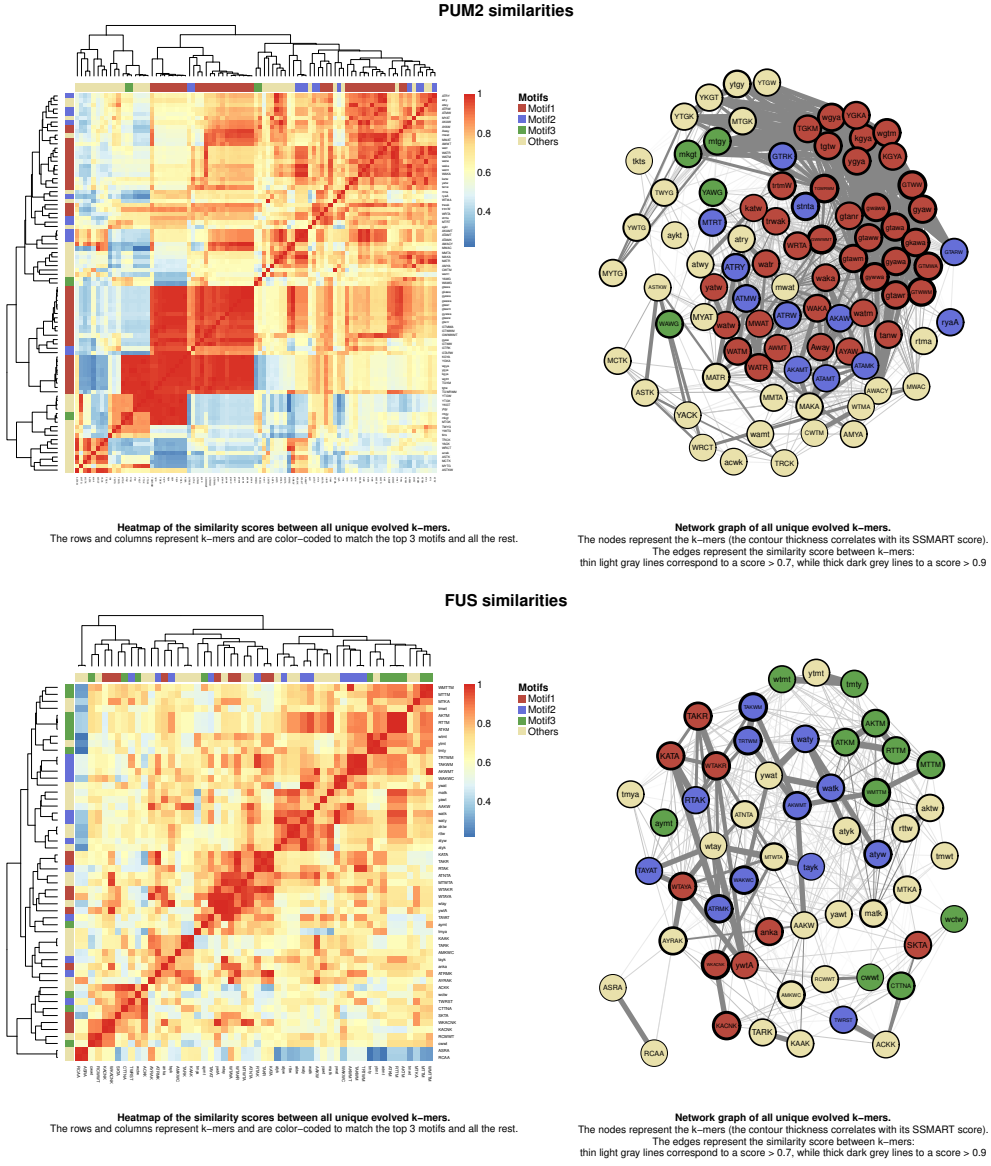


FIGURE 7.4: Visualization of motif clusters. Data corresponds to PUM2 (top) and FUS (bottom) proteins. The panel for each RBP contains a heatmap (left) and a network graph (right). Both plots depict the pair-wise similarities between all unique evolved motifs (k-mers). The k-mers are represented on rows and columns in the heatmap plot and as nodes in the network graph. They are color-coded to match the motif cluster they are assign to in the post-processing step. In the network graph, two motifs are connected by an edge if they have more than 90% similarity (thick dark grey edge) or between 70%-90% similarity (thin light grey line). Similarities bellow 70% are not depicted in this plot.

7.2.3 Synthetic datasets

We generated synthetic datasets that contain specific implanted motifs in various proportions of structured/unstructured binding sites. First, we selected 10 PWMs derived from RNAcompete experiments from the RBP compendium [121]. They all have length 7, but have different nucleotide compositions and their average information content

varies between 0.65 and 1.47 (Table 7.1). We then used a 2nd order Markov chain to generate a large set of 500.000 random 3‘UTR sequences with lengths following the empirical distribution observed in PAR-CLIP peaks. In order to obtain different structural environments, each PWM was implanted into all synthetic 3‘UTR sequences in random locations, and then the secondary structure was predicted. Based on the number of predicted unpaired bases of the implanted motifs, we considered 20 different structural combinations (Fig. 7.5). Structures A-K represent linear combinations of purely single-stranded and double-stranded binding sites, from A with 100% unpaired motifs, to K with 100% paired motifs (with 10% increments). Structures L-Q represent various degrees of double-strandedness in the binding, from set L with all sequences having 1 paired base, to structure Q with 6 paired bases in the implanted motif. The last 3 structural environments (R-T) denote variable structures: R has 30% unpaired motifs and 10% of each set with 1 to 7 paired bases, S has equal numbers (12.5%) of motifs with 0 to 7 paired bases, and T has 35% paired motifs, 35% unpaired motifs and 5% of each of the rest. For each implanted motif and each structural environment, we randomly selected 10 datasets of 2000 sequences each, generating a total of 2.000 synthetic datasets. We then added some noise to this data as follows: we generated a single set of 10000 3‘UTR sequences with the same 2nd order Markov chain, and then we predicted the corresponding secondary structure. In each synthetic dataset, we inserted 500 sequences selected at random from this “noise” set. The resulted datasets represented the “core” data for our comparison.

Since **SSMART** requires as input a binding score for each sequence, we sampled 2500 such scores (conversion specificity) from 25 PAR-CLIP datasets. Then we randomly associated these values to sequences in the generated datasets, making sure that the 500 “noise” sequences will be triangularly distributed among the positive sequences (less at the top, more at the bottom). For *GraphProt*, we generated a “negative” set of 2500 sequences with 3‘UTR composition.

7.2.4 Experimental datasets

We applied **SSMART** on high-throughput *in vivo* and *in vitro* data from CLIP and RNAcompete experiments, respectively. We selected and analysed 10 different proteins: FMR1, FUS, IGF2BP2, IGF2BP3, HuR, LIN28A, QKI, SRSF1, SRSF7 and SRSF9 that have both types of data available. We added two more proteins that had only CLIP data (Table 7.2).

We retrieved the selected RNAcompete datasets from [121]. We downloaded the CLIP datasets from Gene Expression Omnibus [32] and processed the reads as described in

TABLE 7.1: 10 randomly selected PWMs from the RBP compendium. These sequence motifs were used to generate the synthetic datasets used for evaluation.

Motif	Average IC	Consensus	RBP
M159_0.6	1.4693	WGCAUGM	A2BP1, RBFOX2, RBFOX3
M147_0.6	1.3348	GACAGAN	CNOT4
M056_0.6	1.253	ACAACRR	SRSF3
M021_0.6	1.2086	AGGAURA	G3BP2
M232_0.6	1.1782	UUUUUUU	ELAVL1, ELAVL3
M162_0.6	1.0972	AGAAANU	PABPC5
M108_0.6	1.0967	UUUGUUU	ELAVL1, ELAVL3
M242_0.6	1.0365	CCAAAUU	HNRNPR, SYNCRIP
M054_0.6	0.9501	GCGCGCG	RBM8A
M168_0.6	0.6589	GURGUKU	PSPC1, SFPQ

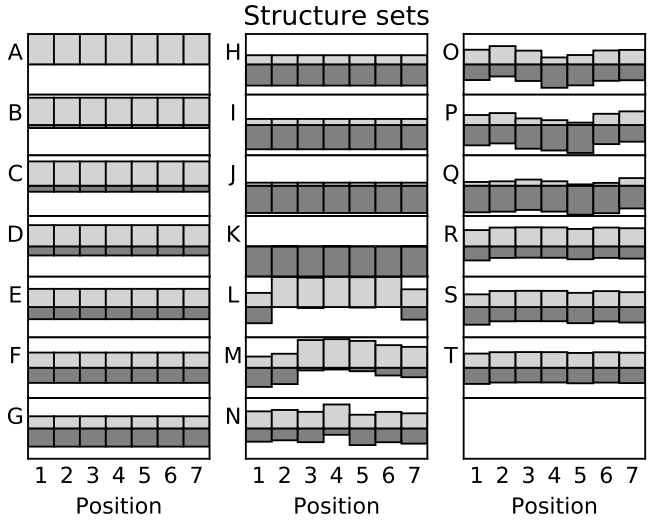


FIGURE 7.5: Structural environments in synthetic data. The overall structure that corresponds to the 10 datasets generated for motif M147 is depicted for each type of structure A-T. For each position in the motif, the light grey rectangle coresponds to the probability of being unpaired, while the dark grey represents the paired probability.

[104]. Briefly, the reads from each library were pre-processed and aligned to the corresponding genome (hg19, mm10) and then interaction sites were defined with PARalyzer [21]. For all PAR-CLIP datasets we considered all PARalyzer clusters that corresponded to mRNAs and to each we associated the T-to-C conversion specificity normalized by gene expression, while in the case of CLIP-seq we used the groups with the normalized read counts. In order to obtain more realistic structure predictions, we extended each binding site by maximum 50 bp on each side using either the genome (for the intronic regions) or the transcriptome (for the rest). For each cell line we derived transcript abundance from RNAseq data, and then for each considered site we retrieved the flanks up to 50 bp from the most abundant transcript that contained it.

Table 7.2: Biological datasets

Protein	CLIP SRA accession number	RNAcompete ID
HuR	SRR189777	RNCMPT00032
QKI	SRR048969	RNCMPT00047
FMR1	SRR527727	RNCMPT00016
LIN28A	SRR531465	RNCMPT00162
LIN28A	SRR458759	
LIN28A	SRR764666	
FUS	SRR070449	RNCMPT00018
PUM2	SRR048968	
ROQUIN	SRR857933	

7.2.5 Evaluation on synthetic datasets

In order to evaluate the motif finders performance we converted all sequence and structure predictions to a uniform encoding. For sequence motifs we used PWMs, converting *RNAcontext* energy matrices and *GraphProt* list of top 1000 sequences to probability matrices. For **SSMART** we colapse the predicted PWM over the 8 letter extended alphabet to a 4 letter alphabet. In the case of structures, *Zagros* and **SSMART** use two structural context per nucleotide (paired and unpaired) while *RNAcontext* and *GraphProt* use larger but distincts sets of structures (e.g. stem, hairpin loop, internal loop, etc.), thus we converted the predicted structures of all tools to a vector of paired probabilities.

We evaluated the motif finders performance on synthetic datasets by computing the recovery rates for sequence and structure motifs separately. For all tools we considered one motif, taking the top one when more motifs are reported. We compared the recovered motifs with the implanted motifs by computing similarity scores based on the Harbison metric (used also in the post-processing step). We use Eq. (7.9) to define the similarity score between two sequence motifs a and b, with the mention that a', b' correspond to all possible overlaps between motifs a, b induced by shifts such that the minimum overlap length is $\max(w-1, 4)$, where w is the smaller motif length. We also define the similarity score for the corresponding structures $s(a)$ and $s(b)$ to be: $sim(s(a), s(b)) = 1 - D(s(a'), (b'))$, where $D(s(a), s(b)) = \frac{1}{w}\sum_{i=1}^{w} |s(a_i) - s(b_i)|$. For each tool, we computed its threshold for "recovered" and "not recovered" motifs by comparing each of the 2000 predicted motifs with all 200 implanted motifs. Then the optimal cutoffs for sequence motifs and for structure motifs are determined independently by optimizing the p-values obtained with G-tests of independence (see Table 7.3 and Fig 7.6) [136].

TABLE 7.3: The cutoffs used to distinguish between "recovered" and "not recovered" motifs on the synthetic data. The tools marked with "-seq" correspond to sequence-only mode.

Motif finder	Sequence	Structure
SSMART	0.8082	0.7188
SSMART-seq	0.8206	0.8725
RNAcontext	0.6995	0.7366
GraphProt	0.6599	0.8232
Zagros	0.817	0.8908

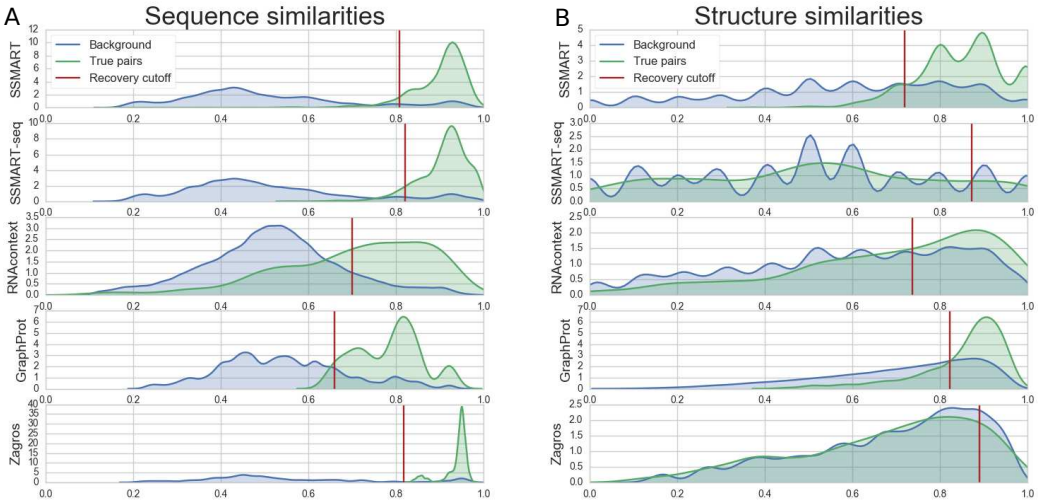


FIGURE 7.6: Distribution of similarity scores between predicted and implanted motifs in synthetic datasets. The scores correspond to sequence motifs (A) and structure motifs (B). The similarities computed for all synthetic sets are depicted in green (True pairs), while the background consisting in the similarities between at possible pairs of predicted and implanted motifs are represented in blue. The vertical brown line represents the optimized cuttof for each tool. All motifs that have scores to the rigth of this line are considered "recovered" by the motif finder.

7.2.6 Evaluation on CLIP datasets

We compared **SSMART**, *GraphProt* and *Zagros* on 6 selected PAR-CLIP libraries corresponding to 2 proteins: ELAVL1 (HuR) and PUM2 (see Table 7.4). For each tool and library we retrieved the predicted sequence motif in the form of a PWM, and the sequence-structure motif as a PWM over the 8 letter extended alphabet. Then we tested how well a motif predicted on a particular library correlates with the binding scores associated with each of the 6 considered libraries. We used the Kendall tau correlation coefficient between the ranked list of binding scores and the corresponding log-likelihood scores of a given PWM. We note that the tau coefficient has values in $[-1, 1]$, a value close to 1 indicating strong agreement, while a value close to -1 indicating strong disagreement. For each tool and protein, we then applied a two-sample Kolmogorov-Smirnov

Table 7.4: CLIP datasets used to compare different motif finders.

Protein	Protocol	Cell line	Reference	SRR accession number	ID
ELAVL1	PAR-CLIP	HEK293	[74]	SRR189777	ELAVL1.1
ELAVL1	PAR-CLIP	HEK293	[103]	SRR248532	ELAVL1.2
ELAVL1	PAR-CLIP	HeLa	[82]	SRR309285	ELAVL1.3A
ELAVL1	PAR-CLIP	HeLa	[82]	SRR309286	ELAVL1.3B
PUM2	PAR-CLIP	HEK293	[53]	SRR048967	PUM2.A
PUM2	PAR-CLIP	HEK293	[53]	SRR048968	PUM2.B

test, comparing the tau correlations on datasets for the same RBP versus those on the other protein.

7.2.7 Parameter optimization

While our tool and *Zagros* do not have parameters that need to be set, *RNAcontext* and *GraphProt* have multiple parameters that influence their performance.

RNAcontext has three important parameters: the motif length w, the structural alphabet e and the number of initializations s. The motif length is specified as a range, and *RNAcontext* uses lerned models for smaller motifs to initialize longer motif lengths. We set w to $4-10$, a range that is consistent with the **SSMART** possible motif lengths. For describing RNA structure, we used the "PHIME" alphabet that consists in five different structures: paired (P), hairpin loop (H), internal loop (I), multiloop (M), and external loop (E). For structure evaluations we considered the paired probabilities. We set parameter s to 3, running the tool with 3 different initializations.

GraphProt has six parameters that can be optimized in a dedicated step, using program option $-ls$. For each motif and type of structure used in the synthetic datasets, we ran the optimization procedure on a separate set of sequences and used the optimized values for all datasets in each category. However, the motif recovery rates with the default parameters were better than those obtained with prior parameter optimization, therefore all results reported here correspond to the default values.

7.3 Results

SSMART performs *de novo* motif discovery on high-throughput RNA-binding protein data, predicting sequence and structure binding motifs of RBPs. We generated synthetic datasets with certain motifs in different structural context in order to evaluate it's performance and to compare the prediction of sequence and structure motifs with three other RBP sequence-structure motif finders: *RNAcontext*, *GraphProt* and *Zagros* [7, 66, 97].

TABLE 7.5: Global recovery rates for sequence and structure motifs on synthetic datasets. The values reported for SSMART-seq correspond to a version of **SSMART** that uses only sequence information.

Tools	SSMART	SSMART-seq	RNAcontext	GraphProt	Zagros
Sequence	91.75	92.05	58.14	94.84	100
Structure	88.65	14.75	50.03	73.25	15.25

We also compared **SSMART** with *GraphProt* and *Zagros* in a cross-validation setting across replicate *in vivo* CLIP libraries. We then used **SSMART** to examine a range of publicly available biological datasets and to compare binding specificities derived from *in vivo* and *in vitro* experiments. Afterwards we analyzed the structural binding specificity for a selection of CLIP datasets. In this section we present the results of our analyses.

7.3.1 Recovering sequence and structure motifs from synthetic datasets

Evaluation of *de novo* motif predictions from experimentally-derived datasets is challenging due to lack a known ground truth and noise. Therefore, we generated a large set of datasets that mimic PAR-CLIP binding sites (clusters) in which we inserted 10 different motifs derived from RNAcompete experiments. For each motif we considered 20 different structural combinations (A-T) and we measured not only how well each tool recovers it, but also how well the initial structure is predicted.

The recovery rates for sequence and structure motifs on all 2000 datasets are presented in Table 7.5. Our tool outperforms all other motif finders in recovering the structure and is outperformed by *GraphProt* and *Zagros* in the sequence recovery, but has the best results for the combination of both sequence and structure. Although *Zagros* recovers 100% of the sequence motifs, its structural predictions are on the same scale as **SSMART**-seq, which considers all motifs to be single-stranded.

The sequence predictions performance is consistent for all tools across different structural environments (Fig 7.7A). We note that the motif appears to have some influence on the prediction performance for some motif finders, for example **SSMART** recovering the sequence motif with the lowest information content in just 46.5% datasets or *GraphProt* recovering the ACAACRR motif in 58% cases. In the case of structure predictions, all the tools exhibit variability across structural environments. **SSMART** and *RNAcontext* perform better on sets with more defined structures, while *GraphProt* recovers the mixed structures. This difference is explained by the way each tool models and reports the motifs. **SSMART** can capture mixed structures by reporting 2 or more separate motifs, but in this settings we consider only the top reported motif. Even so, **SSMART** recovers perfectly the structure if it has 80-100% of the binding sites in either unpaired or paired

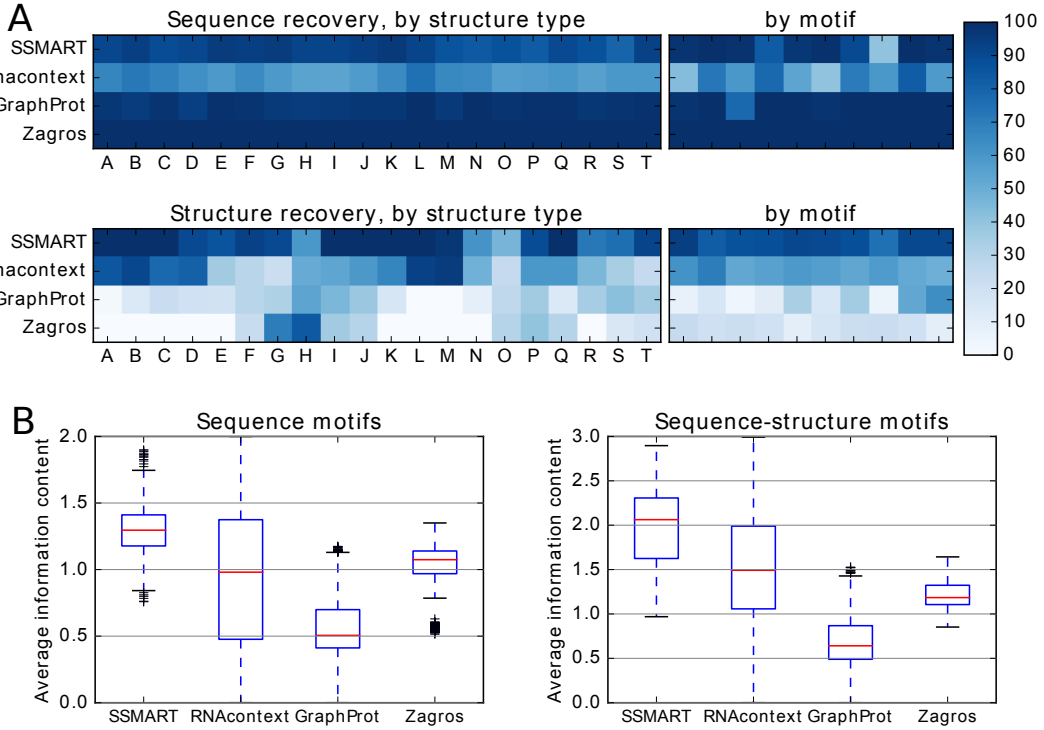


FIGURE 7.7: Results for **SSMART**, *RNAcontext*, *GraphProt* and *Zagros* on synthetic data. (A) Recovery rates for sequence motifs (top) and for structure motifs (bottom). The colors represent the percentage of recovered motifs from the datasets grouped by structure type or by implanted motif. (B) Average information content for predicted motifs either in their sequence component (left) or for combined sequence-structure motifs (right).

states (sets A-C and I-K) or if it has 6 of the 7 bases in the same structural context (sets L and Q). If just 5 bases have the same structure, the recovery rates are 98% and 94% for unpaired and paired RNA, respectively. In summary, **SSMART** recovers more than 90% of structural motifs for 15 (out of 20) structural environments, while *GraphProt* for only 7 types of structures.

Next, to assess the specificity of identified motifs relative to background, we computed the average information content of the predicted motifs (Fig 7.7B). We considered two variants of motif information content: for sequence motifs we used average information content over the 4 letter sequence alphabet $A = \{$ A,C,G,T $\}$; for sequence-structure motifs we derived average information content using the 8 letter sequence-structure alphabet $A_{basic} = \{$A, C, G, T, a, c, g, t$\}$. In the case of sequence motifs, the median information content was 1.29 for **SSMART**, 0.98 for *RNAcontext*, 0.5 for *GraphProt* and 1.07 for *Zagros*. *RNAcontext* sequence motifs cover the whole range of possible information content, while the values for *Zagros* have the smallest variance. The low value for *GraphProt* is explained in part by the length of 12 bases reported for all motifs.

SSMART produces the most expressive sequence-structure motifs, with a median information content of 2.06. The values for *RNAcontext*, *Zagros* and *GraphProt* are 1.49, 1.18 and 0.64 respectively. Taken together these results demonstrate that **SSMART** provides the best all-around performance on the simulated data.

7.3.2 Testing motif predictions on CLIP datasets

Next, we used published CLIP datasets to evaluate the motif predictions of **SSMART**, *GraphProt* and *Zagros* in a train/test setting, by correlating each learned motif with the binding scores of all considered libraries. A meaningful motif will exhibit positive correlation for libraries of the same protein and negative or smaller correlation coefficients for inter-protein tests. The Kendall tau correlation coefficients obtained for 4 ELAVL1 (HuR) and 2 PUM2 PAR-CLIP libraries are presented in Fig 7.8. The datasets denoted with A and B are replicates, while the ones denoted 1, 2, and 3 are from independent experiments. ELAVL1.3 CLIP was performed in HeLa cell line, while the rest in HEK293 cells. For all tools, the sequence motifs trained on the ELAVL1 datasets (first 4 rows) perform as expected, with negative or lower values obtained when tested on PUM2 binding sites, the corresponding p-values being bellow 0.0001 (see Supplementary Table S5). The only outlier is the ELAVL1.3A dataset, on which all tested motifs obtain lower correlations. However, **SSMART** is the only motif finder that shows the same trend not only in the ELAVL1.3A column, but also in the ELAVL1.3A row. On the other hand, the sequence motifs trained on the PUM2 datasets (last 2 rows) are protein-specific only in the case of **SSMART**, with a p-value of 0.002, while for *GraphProt* and *Zagros* the predicted motifs correlate similarly with PUM2 and ELAVL1 binding sites (p-values of 0.335 and 0.061, respectively).

7.3.3 Identification of motifs from *in vivo* and *in vitro* datasets

We applied **SSMART** to 36 CLIP and 21 RNAcompete libraries corresponding to 12 proteins with both *in vivo* and *in vitro* experiments. All RNAcompete data is from human and was downloaded from the compendium of RNA-binding motifs [121]. The *in vivo* data corresponds to 31 PAR-CLIP and 5 CLIP-seq experiments conducted in human HEK293, HeLa, and H9 hESC cells; three of the CLIP-seq datasets were performed in A3 lymphocytes or mESC cells.

First, we compared RBPs for which both *in vivo* and *in vitro* experimental data was available (Fig 7.9). For each RBP we present the top sequence-structure motifs for CLIP and RNAcompete datasets, as well as the motifs reported by the authors of the respective experiments. We note that in the case of RNAcompete experiments, **SSMART** derives

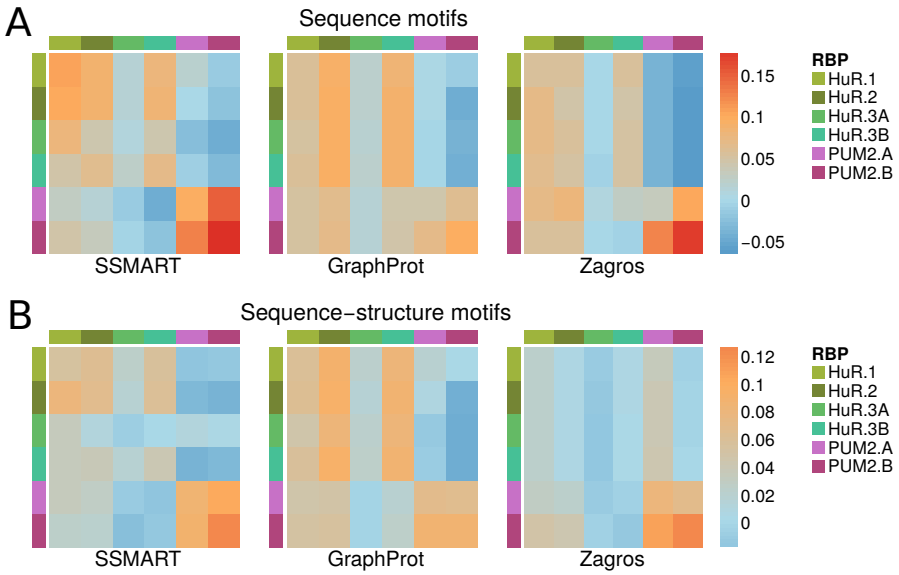


FIGURE 7.8: Results for **SSMART**, *GraphProt* and *Zagros* on biological datasets. Kendall tau correlation coefficients between the motifs predicted on one specific CLIP dataset versus the binding scores of a list of CLIP libraries. The correlations are depicted with the same color scale for the sequence motifs (A) and the combined sequence-structure motifs (B). The rows correspond to the training sets, and the columns to the test sets.

the binding motif from all ≈ 240,000 probes with corresponding affinities, while [121] derived enrichment scores for all possible 7-mers and then defined motifs from the top 10 7-mers. Nevertheless, we find strong agreement between our top predictions and the reported RNAcompete motifs for almost all RBPs examined. None of the *in vitro* results indicated any structural features, which may be due to the length and/or selection of RNA oligos in the RNAcompete assay.

RBPs exhibited varying degrees of concordance between A) predictions and results for *in vitro* data, B) predictions and reported results for *in vivo* data, and C) predictions and results for *in vivo* and *in vitro* data. For both ELAVL1 (HuR) and QKI, we observed full agreement (i.e. *in vivo* and *in vitro* predictions and results identified the same motif). For ELAVL1, the U-rich sequence motif recovered by **SSMART** from *in vivo* data was associated with mostly single-stranded structural context. As expected [39], the motifs reported for QKI were associated with single-stranded structural context.

Fragile X-mental retardation 1 (FMR1) is a RNA-binding protein that has multiple distinct RNA-binding domains. PAR-CLIP experiments reported two short binding motifs, ACUK and WGGA, that interact with the KH and RGG domains, respectively [4]. Our top predicted motif is similar to WGGA and associated with paired RNA, which may reflect previously reported binding to G-quadreplex structures [16]. A CU di-nucleotide, which is present in the secondary ACUK motif, was present in the top

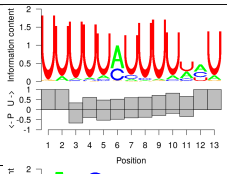
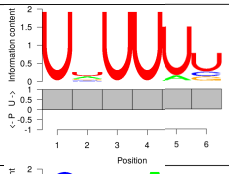
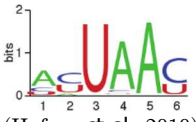
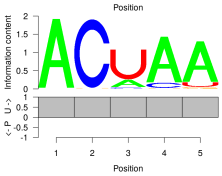
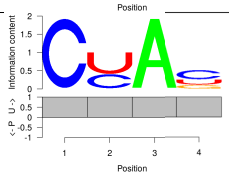
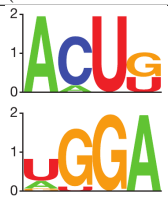
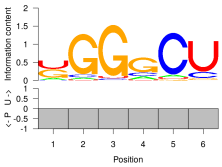
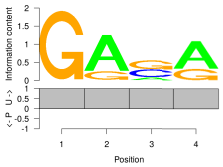
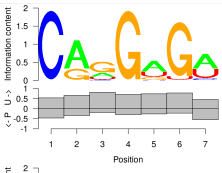
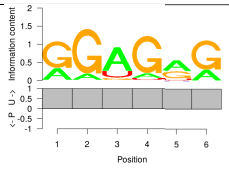
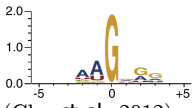
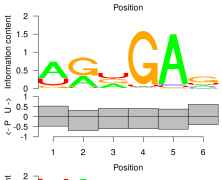
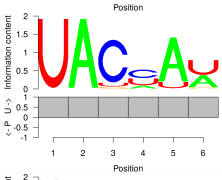
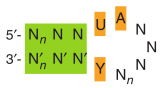
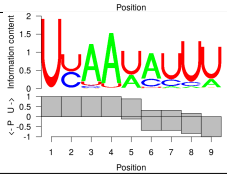
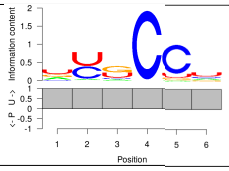
Protein	CLIP datasets: Reported motif	CLIP datasets: SSMART motif	RNAcompete datasets: SSMART motif	RNAcompete datasets: Reported motif
HuR	binds low-complexity, uracil (U)-rich elements (Kishore et al. 2011)			 (Ray et al. 2013)
QKI	 (Hafner et al. 2010)			 (Ray et al. 2013)
FMR1	 (Ascano et al. 2012)			 (Ray et al. 2013)
LIN28A	 (Wilbert et al. 2012)			 (Ray et al. 2013)
	 (Cho et al. 2012)			
	AYYHY (Y = U,C and H= A,C,U) (Hafner et al. 2013)			
FUS	 (Hoell et al. 2011)			 (Ray et al. 2013)

FIGURE 7.9: **SSMART** results on biological data: *in vivo* vs. *in vitro* comparison.

scoring motif. The *in vitro* predicted and reported results did not identify the secondary motif bound by the KH domain, nor did it indicate a structural context for the WGGA.

The *in vitro* prediction for lin-28 homolog A (LIN28A) was consistent with the reported motif from RNAcompete. Similarly, our *in vivo* predictions from LIN28 CLIP-seq data from Yeo and Kim labs identified a GA-rich motif consistent with what was reported [19, 151]. Both the predicted and reported results from LIN28A PAR-CLIP were not consistent with the *in vitro* results or the *in vivo* CLIP results. Also, for some analyzed proteins the *in vitro* predictions were in agreement with the reported compendium motif, but the predicted motifs for *in vivo* datasets showed different binding specificities (data

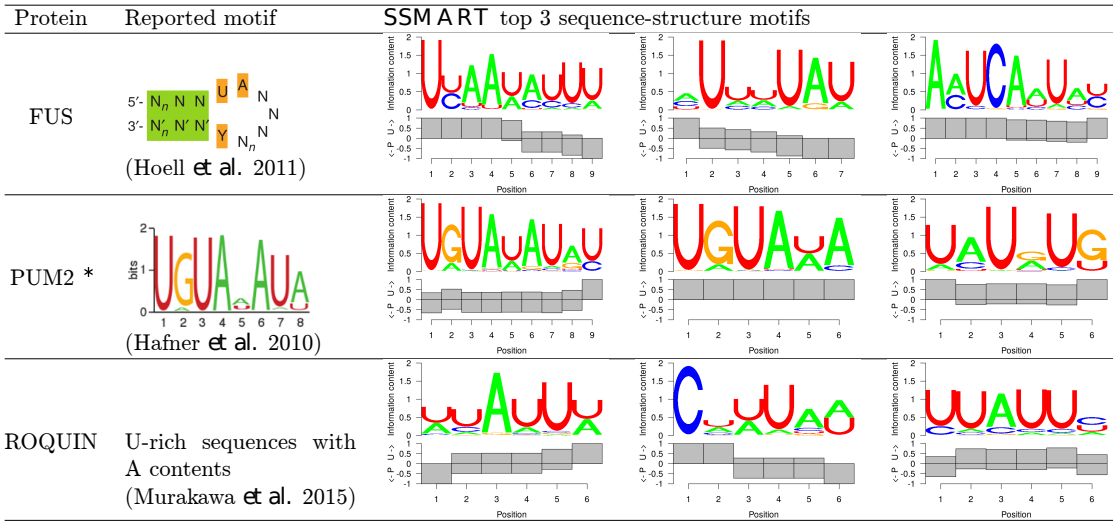


FIGURE 7.10: **SSMART** top 3 sequence-structure motifs for three selected CLIP datasets. * marks predictions obtained with the random set scoring

not shown). The basis for the inconsistency is unclear and could be due to technical differences between CLIP and PAR-CLIP, ranking and normalization of called peaks, or the cell lines the experiments were performed in.

FUS was a clear case in which there was concordance between our predictions and reported results both *in vivo* and *in vitro*, however the reported *in vivo* and *in vitro* specificities differ. The *in vitro* results indicate a CG-rich motif, while the *in vivo* results suggest a UA-rich sequence, which has been reported to have structural context [60], which we will describe in more detail below. This may represent an example in which the *in vitro* results may not accurately reflect *in vivo* binding.

Examining the structural context of binding specificity

Due to the lack of structural insights from the RNAcompete results, we focused on *in vivo* predictions exhibiting markedly different structural context for a subset of RBPs (Fig 7.10). A stem-loop structure with some sequence preference was previously reported for FUS [60]. The top ranked motif predicted for the FUS PAR-CLIP data was consistent with the reported binding preference in both sequence and structure. From 5' to 3' the predicted motif is decreasingly single-stranded particularly with an apparent transition from unpaired to paired at position 6, presumably representing the loop, with the reported UA at the begining of the loop (positions 4-5).

Examination of Pum2 PAR-CLIP data revealed the well-established UGUAHAUA binding motif [53]. As expected, we found this motif in a single-stranded context [91]. Interestingly, we also identified the motif in a paired context, which may represent sites in

which modulation of secondary structural switch influences Pum and miRNA-mediated regulation [67].

Examination of Roquin (RC3H1) binding specificity using PAR-CLIP did not reveal a specific sequence motif, however they proposed existence of a stem-loop with an AU-rich loop region [108]. The top predicted motif match is consistent with the reported sequence description. This prediction could represent, predominantly, the loop portion of the proposed stem-loop. The results described for these three RBPs highlight the manner in which incorporation of secondary structure can enhance the interpretation of RBP-binding specificity, particularly for *in vivo* experimental data.

7.4 Discussion[3]

Identifying the sequence and structure recognition motifs of RBPs is essential for understanding eukaryotic gene regulation. Due to advances in high-throughput technologies for measuring RBP-DNA binding both *in vivo* and *in vitro* (such as CLIP assays and RNAcompete), it is now possible to identify both transcriptome-wide targets of RBPs and their intrinsic RBP-DNA binding specificity. Despite the fact that RNA binding proteins achieve their binding specificity through a combination of sequence and structure recognition, for most RBPs only a primary sequence motif has been determined, while the structure of the binding sites remains largely uncharacterized. We developed **SSMART**, a *de novo* motif finder that identifies sequence-structure binding motifs from large sets of RNA sequences derived from genome-wide *in vivo* or *in vitro* experiments. Our tool simultaneously models the primary sequence and the structural properties of the RNA target sites and produces easy to interpret sequence-structure binding motifs. **SSMART** searches for optimal sequence-structure motifs of flexible length in putative RBP binding sites ranked by their experimentally-derived binding evidence. While *Zagros* and *GraphProt* were designed for CLIP data and *RNAcontext* is best suited for RNAcompete data, our approach is more general and can successfully handle different types of input. Moreover, **SSMART** learns all motif characteristics from the data, including the motif length, and does not require parameter optimization. Like *Zagros*, **SSMART** accounts only for double-stranded and single-stranded preferences at each individual position of the motif. In contrast, *RNAcontext* and *GraphProt* distinguish between five different structural contexts, but they output aggregate structural motifs. Our tool is able to identify different sequence motifs with the corresponding per base structural preference for the same protein.

[3]This section was published as [106]

Although it was reassuring that **SSMART** performed well on simulated and *in vitro* data, ranking binding evidence is straightforward in these scenarios, unlike for *in vivo* binding sites. RNA expression levels clearly impact the read-evidence and there are other factors, such as cross-linking or RNase choice, that may need to be incorporated to properly rank *in vivo* binding sites. Therefore input or background binding libraries may be more useful, particularly for intronic binding sites for which RNA expression estimates could be problematic. Appropriate normalization and ranking of *in vivo* (i)CLIP or PAR-CLIP data remains an ongoing challenge in the field.

We identified cases in which *in vitro* and *in vivo* results were discordant. Biases in both *in vitro* and *in vivo* assays may explain these differences. However, these differences could also be due to factors influencing *in vivo* binding that cannot be recapitulated *in vitro*. Biologically relevant explanations include RNA structural constraints, multiprotein RNA-binding complexes, as well as biophysical features of RNP granules in which these interactions occur. Our results indicate that **SSMART** should assist investigators in accounting for RNA-structural constraints. Importantly, **SSMART** is general enough to be utilized as the determination of RNA-structure progress both experimentally and computationally.

In conclusion, we propose an efficient algorithm to identify the most probable sequence-structure motif, or combination of motifs, given a large set of RNA sequences. Our method can contribute to the systematic understanding of RBP-RNA binding specificity as more genome-wide experiments that determine RBP binding are performed.

Chapter 8

Conclusions and outlook

This work focused on the specificity of interactions between regulatory proteins and nucleic acids, resulting in two main computational advances: (1) **COUGER** [105, 107], a classification framework that distinguishes between TF-DNA interactions for TFs with similar binding motifs (Chapter 6) and (2) **SSMART** [106], a tool for *de novo* identification of the sequence-structure binding motifs for RBP-RNA interactions (Chapter 7). This last chapter summarizes some of the general conclusions and outlines a number of possible future directions for the two related topics.

8.1 Distinguishing between TF-DNA interactions for TFs with similar binding motifs

Most eukaryotic TFs are part of large protein families and despite the fact that multiple paralogous TFs are co-expressed in the cell, little is known about how these proteins achieve their binding specificity. We showed that information about putative co-factors binding preferences can be successfully used to classify between genomic regions bound *in vivo* by paralogous TFs. Our classification approach, **COUGER**, performed consistently better than chance. When applied on factor specific ChIP-seq peaks for 20 pairs of related TFs, with features derived from PBM or PWM data, the median accuracy, precision, sensitivity and specificity were 85.7%, 85.9%, 84.1% and 86.1%, respectively. However, the predicted sets of co-factors need to be further validated, one question being if they are bound *in vivo* to TF1- or TF2-unique sequences. This aspect can be addressed by using also ChIP-seq datasets to derive the putative co-factors classification features. Such a framework can incorporate ChIP-seq information not only for TFs, but also features reflecting the histone modification and chromatin-modifying factors. Frank *et al.* [44], for example, combined transcription factors and histone marks in a random

forest classifier to discriminate between PU.1-bound DNase I-hypersensitive sites that open or remain stable with a certain treatment.

Another possible approach for studying the specificity of TF paralogs is to integrate their *in vitro* PBM binding affinities with the *in vivo* ChIP-seq binding sites. Shen *et al.* present a framework called iMADS (integrative Modeling and Analysis of Differential Specificity) that uses *in vitro* data to model and predict differential *in vivo* binding [134]. Their proposed method uses as input high-throughput quantitative PBM data for a pair of paralogous TFs of interest, as well as biological replicates. One component uses support vector regression to model the DNA-binding specificity for each individual TF, and the other one uses weighted least squares regression to model the differential specificity between them. When applied to a selected set of 11 paralogous TFs from 4 distinct human TF families, iMADS finds that paralogous TFs show some different intrinsic preferences for DNA, most of them being in the medium and low affinity ranges. These differences can partially explain differential genomic binding. Therefore one can use the iMADS tool to identify the *in vivo* differences that are explained by the *in vitro* specificity of TF paralogs, and then apply **COUGER** to the rest of the DNA sequences. The two classes used in **COUGER** could also be refined using other types of adjacent information, but this path focuses on the pre-processing steps. An alternative approach is to add features that integrate the *in vitro* specificity differences.

A different direction for **COUGER** is to apply the classification framework to RNA-binding proteins. Many RBPs are also part of protein families and have similar binding preferences. With the characterization of a large set of RBP sequence motifs [121], such an approach is now feasible.

8.2 Identifying the binding motif for RBP-RNA interactions

The RBPs bind to RNA molecules by recognizing a combination of sequence and structure properties in variable proportions. For a long time, the research focused on determining the sequence motifs for these regulatory proteins, as in the case of TFs, and the structural motifs for many RBPs remained uncharacterized. Although in recent years some tools were designed to account for the RNA secondary structure when modeling the binding preferences of RBPs, there is room for improvement. We introduced **SSMART**, a *de novo* motif finder that simultaneously models the primary sequence and the secondary structure of the RNA. When compared to existing tools in thorough benchmarking using simulated datasets with varying amounts of sequence and structure content, **SSMART** performed favorably recovering the implanted sequence motifs

in 95.8% of the datasets, and the structure motifs in 74.6% cases. Our tool recovered also the known sequence motifs from high-throughput *in vitro* and *in vivo* RBP-RNA interaction datasets. If the application of **SSMART** on *in vitro* RNAcompete datasets is straightforward, the use of *in vivo* binding sites is more challenging due to the need of a quantitative score for the binding strength. Even if we tested multiple scores for PAR-CLIP and CLIP-seq data, none of the variants resulted in better predictions across all tested datasets. For both types of experiments, we selected scores normalized by the transcript abundancy (measured with RNA-seq), that recovered best the known motifs for most of the considered RBPs. **SSMART** would benefit from a better defined score for *in vivo* binding.

Another improvement related to the input data concerns the RNA secondary structure information used. The topic of accurately and efficiently folding a stretch of RNA nucleotides is still open. As better structural predictions are made or more experimental measurements become available, the derived sequence-structure motifs will become more accurate. Even so, one limitation of our model comes from the small number of possible values for the structural states, namely paired or unpaired. In order to include multiple states for the ssRNA, a possible approach is to use a probabilistic model for the sequence and structure binding preferences of RBPs. For such a model a sampling approach may be more appropriate.

A promising direction for identifying the binding motifs for RBP-RNA interactions consist in using deep learning approaches and modeling the sequence and/or the structure motifs across multiple RBPs simultaneously. For example, Alipanahi *et al.* proposed DeepBind, a tool based on deep learning for predicting the sequence motifs of regulatory proteins [2].

Bibliography

[1] Alberts, B., Johnson, A., *et al.* (2002). *Molecular biology of the cell*. Garland Science.

[2] Alipanahi, B., Delong, A., *et al.* (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–8.

[3] Anders, G., Mackowiak, S. D., *et al.* (2012). doRiNA: a database of RNA interactions in post-transcriptional regulation. *Nucleic acids research*, 40(Database issue):D180–6.

[4] Ascano, M., Mukherjee, N., *et al.* (2012). FMRP targets distinct mRNA sequence elements to regulate protein expression. *Nature*, 492(7429):382–6.

[5] Avery, O. T. (1944). Studies on the Chemical Nature of the Substance Inducing Transformation of Pneumococcal Types: Induction of Transformation By a Desoxyribonucleic Acid Fraction Isolated From Pneumococcus Type Iii. *Journal of Experimental Medicine*, 79(2):137–158.

[6] Badis, G., Berger, M. F., *et al.* (2009). Diversity and Complexity in DNA Recognition by Transcription Factors. *Science*, 324(5935):1720–1723.

[7] Bahrami-Samani, E., Penalva, L. O. F., *et al.* (2015). Leveraging cross-link modification events in CLIP-seq for motif discovery. *Nucleic acids research*, 43(1):95–103.

[8] Bailey, T. L., Williams, N., *et al.* (2006). MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic acids research*, 34(Web Server issue):W369–73.

[9] Baltz, A., Munschauer, M., *et al.* (2012). The mRNA-Bound Proteome and Its Global Occupancy Profile on Protein-Coding Transcripts. *Molecular Cell*, 46(5):674–690.

[10] Bassuk, A. G. and Leiden, J. M. (1995). A direct physical association between ETS and AP-1 transcription factors in normal human T cells. *Immunity*, 3(2):223–37.

[11] Berger, M. F., Philippakis, A. A., *et al.* (2006). Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nat. Biotechnol.*, 24(11):1429–1435.

[12] Bernhart, S. H., Hofacker, I. L., and Stadler, P. F. (2006). Local RNA base pairing probabilities in large sequences. *Bioinformatics*, 22(5):614–5.

[13] Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 24(2):123–140.

[14] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[15] Breiman, L., Friedman, J., *et al.* (1984). *Classification and regression trees*. Wadsworth & Brooks.

[16] Brown, V., Jin, P., *et al.* (2001). Microarray Identification of FMRP-Associated Brain mRNAs and Altered mRNA Translational Profiles in Fragile X Syndrome. *Cell*, 107(4):477–487.

[17] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol*, 2:27:1–27:27.

[18] Chen, G. and Zhou, Q. (2011). Searching chip-seq genomic islands for combinatorial regulatory codes in mouse embryonic stem cells. *BMC Genomics*, 12(1):515.

[19] Cho, J., Chang, H., *et al.* (2012). LIN28A is a suppressor of ER-associated translation in embryonic stem cells. *Cell*, 151(4):765–77.

[20] Cook, K. B., Hughes, T. R., and Morris, Q. D. (2014). High-throughput characterization of protein-RNA interactions. *Briefings in Functional Genomics*, 14(1):74–89.

[21] Corcoran, D. L., Georgiev, S., *et al.* (2011). PARalyzer : definition of RNA binding sites from PAR-CLIP short-read sequence data. *Genome Biology*, 12(8):R79.

[22] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

[23] Crick, F. H. (1958). On protein synthesis. *Symposia of the Society for Experimental Biology*, 12:138–63.

[24] Darnell, R. B. (2010). HITS-CLIP: panoramic views of protein-RNA regulation in living cells. *Wiley interdisciplinary reviews. RNA*, 1(2):266–86.

[25] Delisi, C. and Crothers, D. M. (1971). Prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America*, 68(11):2682–5.

[26] Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, 1857:1–15.

[27] Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comp. Bio.*, 3(2):185–205.

[28] Ding, Y. and Lawrence, C. E. (2003). A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic acids research*, 31(24):7280–7301.

[29] Dobin, A., Davis, C. A., *et al.* (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21.

[30] Doshi, K. J., Cannone, J. J., *et al.* (2004). Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. *BMC bioinformatics*, 5:105.

[31] Eddy, S. R. and Durbin, R. (1994). RNA sequence analysis using covariance models. *Nucleic acids research*, 22(11):2079–88.

[32] Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–10.

[33] Ellington, A. D. and Szostak, J. W. (1990). In vitro selection of RNA molecules that bind specific ligands. *Nature*, 346(6287):818–22.

[34] ENCODE Project Consortium (2011). A user's guide to the encyclopedia of DNA elements (ENCODE). *PLoS biology*, 9(4):e1001046.

[35] ENCODE Project Consortium, Bernstein, B. E., *et al.* (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74.

[36] Estévez, P. a., Tesmer, M., *et al.* (2009). Normalized mutual information feature selection. *IEEE Trans. Neural Netw.*, 20(2):189–201.

[37] Farsani, S. and Mahdavi, M. (2011). Quantification of Gene Expression Based on Microarray Experiment. In *Bioinformatics - Trends and Methodologies*. InTech.

[38] Felsenfeld, G. and Rich, A. (1957). Studies on the formation of two- and three-stranded polyribonucleotides. *Biochimica et Biophysica Acta*, 26(3):457–468.

[39] Feracci, M., Foot, J. N., *et al.* (2016). Structural basis of RNA recognition and dimerization by the STAR proteins T-STAR and Sam68. *Nature Communications*, 7:10355.

[40] Foat, B. C., Morozov, A. V., and Bussemaker, H. J. (2006). Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. *Bioinformatics*, 22(14):e141–9.

[41] Foat, B. C. and Stormo, G. D. (2009). Discovering structural cis-regulatory elements by modeling the behaviors of mRNAs. *Molecular systems biology*, 5:268.

[42] Fong, A. P., Yao, Z., *et al.* (2012). Genetic and epigenetic determinants of neurogenesis and myogenesis. *Dev. Cell*, 22(4):721–35.

[43] Fonseca, N. A., Rung, J., *et al.* (2012). Tools for mapping high-throughput sequencing data. *Bioinformatics*, 28(24):3169–3177.

[44] Frank, C. L., Manandhar, D., *et al.* (2016). HDAC inhibitors cause site-specific chromatin remodeling at PU.1-bound enhancers in K562 cells. *Epigenetics & Chromatin*, 9(1):15.

[45] Freund, Y. and Schapire, R. E. (1996). Experiments with a New Boosting Algorithm.

[46] Georgiev, S., Boyle, A. P., *et al.* (2010). Evidence-ranked motif identification. *Genome Biology*, 11(2):R19.

[47] Gerstein, M. B., Kundaje, A., *et al.* (2012). Architecture of the human regulatory network derived from ENCODE data. *Nature*, 489(7414):91–100.

[48] Gordân, R., Hartemink, A. J., and Bulyk, M. L. (2009). Distinguishing direct versus indirect transcription factor–DNA interactions. *Genome Research*, 19(11):2090–2100.

[49] Gordân, R., Murphy, K., *et al.* (2011). Curated collection of yeast transcription factor DNA binding specificity data reveals novel structural and gene regulatory insights. *Genome Biology*, 12(12):R125+.

[50] Gordân, R., Shen, N., *et al.* (2013). Genomic Regions Flanking E-Box Binding Sites Influence DNA Binding Specificity of bHLH Transcription Factors through DNA Shape. *Cell Rep.*, 3(4):1093–1104.

[51] Graf, R., Munschauer, M., *et al.* (2013). Identification of LIN28B-bound mRNAs reveals features of target recognition and regulation. *RNA biology*, 10(7):1146–59.

[52] Griffiths-Jones, S., Moxon, S., *et al.* (2005). Rfam: annotating non-coding RNAs in complete genomes. *Nucleic acids research*, 33(Database issue):D121–4.

[53] Hafner, M., Landthaler, M., *et al.* (2010). Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, 141(1):129–41.

[54] Hafner, M., Max, K. E. A., *et al.* (2013). Identification of mRNAs bound and regulated by human LIN28 proteins and molecular requirements for RNA recognition. *RNA*, 19(5):613–26.

[55] Harbison, C. T., Gordon, D. B., *et al.* (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104.

[56] Herzig, T. C., Jobe, S. M., *et al.* (1997). Angiotensin II type1a receptor gene expression in the heart: AP-1 and GATA-4 participate in the response to pressure overload. *Proc. Natl. Acad. Sci. USA*, 94(14):7543–8.

[57] Heyne, S., Costa, F., *et al.* (2012). GraphClust: alignment-free structural clustering of local RNA secondary structures. *Bioinformatics*, 28(12):i224–32.

[58] Hiller, M., Pudimat, R., *et al.* (2006). Using RNA secondary structures to guide sequence motif finding towards single-stranded regions. *Nucleic acids research*, 34(17):e117.

[59] Hinegradner, R. T. and Engelberg, J. (1963). Rational for a Universal Genetic Code. *Science*, 142(3595):1083–1085.

[60] Hoell, J. I., Larsson, E., *et al.* (2011). RNA targets of wild-type and mutant FET family proteins. *Nature structural & molecular biology*, 18(12):1428–31.

[61] Ibrahim, M. M., Lacadie, S. A., and Ohler, U. (2014). JAMM: A Peak Finder for Joint Analysis of NGS Replicates. *Bioinformatics*, 31(1):btu568–.

[62] Johnson, D. S., Mortazavi, A., *et al.* (2007). Genome-Wide Mapping of in Vivo Protein-DNA Interactions. *Science*, 316(5830):1497–1502.

[63] Jolma, A., Yan, J., *et al.* (2013). DNA-binding specificities of human transcription factors. *Cell*, 152(1-2):327–39.

[64] Kawana, M., Lee, M. E., *et al.* (1995). Cooperative interaction of GATA-2 and AP1 regulates transcription of the endothelin-1 gene. *Mol. Cell. Biol.*, 15(8):4225–31.

[65] Kazan, H. and Morris, Q. (2013). RBPmotif: a web server for the discovery of sequence and structure preferences of RNA-binding proteins. *Nucleic acids research*, 41(Web Server issue):W180–6.

[66] Kazan, H., Ray, D., *et al.* (2010). RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins. *PLoS computational biology*, 6:e1000832.

[67] Kedde, M., van Kouwenhove, M., *et al.* (2010). A Pumilio-induced RNA structure switch in p27-3 UTR controls miR-221 and miR-222 accessibility. *Nature Cell Biology*, 12(10):1014–1020.

[68] Keene, J. D. (2007). RNA regulons: coordination of post-transcriptional events. *Nature Reviews Genetics*, 8(7):533–543.

[69] Kharchenko, P. V., Tolstorukov, M. Y., and Park, P. J. (2008). Design and analysis of ChIP-seq experiments for DNA-binding proteins. *Nat. Biotechnol.*, 26(12):1351–1359.

[70] Khorshid, M., Rodak, C., and Zavolan, M. (2011). CLIPZ: a database and analysis environment for experimentally determined binding sites of RNA-binding proteins. *Nucleic acids research*, 39(Database issue):D245–52.

[71] Kim, D., Pertea, G., *et al.* (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology*, 14(4):R36.

[72] Kim, D.-K. and Lee, Y.-M. (2004). Requirement of c-jun transcription factor on the mouse mast cell protease-6 expression in the mast cells. *Arch. Biochem. Biophys.*, 431(1):71–8.

[73] Kiryu, H., Terai, G., *et al.* (2011). A detailed investigation of accessibilities around target sites of siRNAs and miRNAs. *Bioinformatics*, 27(13):1788–1797.

[74] Kishore, S., Jaskiewicz, L., *et al.* (2011). A quantitative analysis of CLIP methods for identifying binding sites of RNA-binding proteins. *Nature Methods*, 8(7):559–564.

[75] König, J., Zarnack, K., *et al.* (2011). Protein-RNA interactions: new genomic technologies and perspectives. *Nature reviews. Genetics*, 13(2):77–83.

[76] Konig, J., Zarnack, K., *et al.* (2010). iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. *Nature structural & molecular biology*, 17(7):909–15.

[77] Landt, S. G., Marinov, G. K., *et al.* (2012). ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Res.*, 22(9):1813–31.

[78] Lange, S. J., Maticzka, D., *et al.* (2012). Global or local? Predicting secondary structure and accessibility in mRNAs. *Nucleic acids research*, 40(12):5215–26.

[79] Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359.

[80] Langmead, B., Trapnell, C., *et al.* (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, 10(3):R25.

[81] Latchman, D. S. (2005). *Gene regulation : a eukaryotic perspective*. Taylor & Francis.

[82] Lebedeva, S., Jens, M., *et al.* (2011). Transcriptome-wide analysis of regulatory interactions of the RNA-binding protein HuR. *Molecular cell*, 43(3):340–52.

[83] Lekprasert, P., Mayhew, M., and Ohler, U. (2011). Assessing the utility of thermodynamic features for microRNA target prediction under relaxed seed and no conservation requirements. *PloS one*, 6(6):e20622.

[84] Li, B. and Dewey, C. N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12(1):323.

[85] Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760.

[86] Li, Q., Brown, J. B., *et al.* (2011). Measuring reproducibility of high-throughput experiments. *Ann. Appl. Stat.*, 5(3):1752–1779.

[87] Li, X., Kazan, H., *et al.* (2014). Finding the target sites of RNA-binding proteins. *Wiley interdisciplinary reviews. RNA*, 5(1):111–30.

[88] Li, X., Quon, G., *et al.* (2010). Predicting in vivo binding sites of RNA-binding proteins using mRNA secondary structure. *RNA*, 16(6):1096–107.

[89] Lin, C. Y., Lovén, J., *et al.* (2012). Transcriptional Amplification in Tumor Cells with Elevated c-Myc. *Cell*, 151(1):56–67.

[90] Lorenz, R., Bernhart, S. H., *et al.* (2011). ViennaRNA Package 2.0. *Algorithms for molecular biology*, 6:26.

[91] Lu, G. and Hall, T. (2011). Alternate Modes of Cognate RNA Recognition by Human PUMILIO Proteins. *Structure*, 19(3):361–367.

[92] Lukong, K. E., Chang, K.-w., *et al.* (2008). RNA-binding proteins in human genetic disease. *Trends in genetics : TIG*, 24(8):416–25.

[93] Lunde, B. M., Moore, C., and Varani, G. (2007). RNA-binding proteins: modular design for efficient function. *Nature reviews. Molecular cell biology*, 8(6):479–90.

[94] Machanick, P. and Bailey, T. L. (2011). MEME-ChIP: motif analysis of large DNA datasets. *Bioinformatics*, 27(12):1696–1697.

[95] Marin, R. M. and Vanicek, J. (2011). Efficient use of accessibility in microRNA target prediction. *Nucleic Acids Research*, 39(1):19–29.

[96] Mathews, D. H., Sabina, J., *et al.* (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288(5):911–940.

[97] Maticzka, D., Lange, S. J., *et al.* (2014). GraphProt: modeling binding preferences of RNA-binding proteins. *Genome Biology*, 15(1):R17.

[98] Matys, V., Kel-Margoulis, O. V., *et al.* (2006). TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res.*, 34(Database issue):D108–D110.

[99] McDowall, M. D., Scott, M. S., and Barton, G. J. (2009). PIPs: human protein-protein interaction prediction database. *Nucleic Acids Res.*, 37(Database issue):D651–6.

[100] McLeay, R. and Bailey, T. (2010). Motif Enrichment Analysis: a unified framework and an evaluation on ChIP data. *BMC Bioinformatics*, 11(1):165+.

[101] Mordelet, F., Horton, J., *et al.* (2013). Stability selection for regression-based models of transcription factor-DNA binding specificity. *Bioinformatics*, 29(13):i117–25.

[102] Morgan, T. H. (1911). Random Segregation Versus Coupling in Mendelian Inheritance. *Science*, 34(873):384–384.

[103] Mukherjee, N., Corcoran, D. L., *et al.* (2011). Integrative regulatory mapping indicates that the RNA-binding protein HuR couples pre-mRNA processing and mRNA stability. *Molecular cell*, 43(3):327–39.

[104] Mukherjee, N., Jacobs, N. C., *et al.* (2014). Global target mRNA specification and regulation by the RNA-binding protein ZFP36. *Genome Biology*, 15(1):R12.

[105] Munteanu, A. and Gordân, R. (2013). Distinguishing between genomic regions bound by paralogous transcription factors. In Deng, M., Jiang, R., *et al.*, editors, *Res. Comput. Mol. Bio.*, volume 7821 of *Lect. Notes Comput. Sc.*, pages 145–157. Springer Berlin Heidelberg.

[106] Munteanu, A., Mukherjee, N., and Ohler, U. (2018). SSMART: Sequence-structure motif identification for RNA-binding proteins. *bioRxiv*.

[107] Munteanu, A., Ohler, U., and Gordân, R. (2014). COUGER-co-factors associated with uniquely-bound genomic regions. *Nucleic acids research*, 0(0):gku435–.

[108] Murakawa, Y., Hinz, M., *et al.* (2015). RC3H1 post-transcriptionally regulates A20 mRNA and modulates the activity of the IKK/NF-κB pathway. *Nature Communications*, 6:7367.

[109] Nirenberg, M. W., Jones, O. W., *et al.* (1963). On the Coding of Genetic Information. *Cold Spring Harbor Symposia on Quantitative Biology*, 28(0):549–557.

[110] Nussinov, R. and Jacobson, A. B. (1980). Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Sciences of the United States of America*, 77(11):6309–13.

[111] Oostra, B. A. and Willemsen, R. (2009). FMR1: a gene with three faces. *Biochimica et biophysica acta*, 1790(6):467–77.

[112] Pan, Q., Shai, O., *et al.* (2008). Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature Genetics*, 40(12):1413–1415.

[113] Park, D., Lee, Y., *et al.* (2013). Widespread Misinterpretable ChIP-seq Bias in Yeast. *PloS One*, 8(12):e83506.

[114] Pearson, H. (2006). What is a gene? *Nature*, 441(7092):398–401.

[115] Peng, H., Long, F., and Ding, C. (2005). Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238.

[116] Portales-Casamar, E., Thongjuea, S., *et al.* (2010). JASPAR 2010: the greatly expanded open-access database of transcription factor binding profiles. *Nucleic Acids Res.*, 38(suppl 1):D105–D110.

[117] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

[118] Quinlan, J. R. J. R. (1993). *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers.

[119] Rabani, M., Kertesz, M., and Segal, E. (2011). Computational prediction of RNA structural motifs involved in post-transcriptional regulatory processes. *Methods in molecular biology*, 714:467–79.

[120] Ray, D., Kazan, H., *et al.* (2009). Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nature biotechnology*, 27(7):667–70.

[121] Ray, D., Kazan, H., *et al.* (2013). A compendium of RNA-binding motifs for decoding gene regulation. *Nature*, 499(7457):172–7.

[122] Re, A., Joshi, T., *et al.* (2014). Rna-protein interactions: An overview. In Gorodkin, J. and Ruzzo, W. L., editors, *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods*. Springer.

[123] Ren, B., Robert, F., *et al.* (2000). Genome-wide location and function of DNA binding proteins. *Science (New York, N.Y.)*, 290(5500):2306–2309.

[124] Ricci, E. P., Kucukural, A., *et al.* (2014). Staufen1 senses overall transcript secondary structure to regulate translation. *Nature structural & molecular biology*, 21(1):26–35.

[125] Rich, A. (2009). The Era of RNA Awakening: Structural biology of RNA in the early years. *Quarterly Reviews of Biophysics*, 42(02):117.

[126] Riordan, D. P., Herschlag, D., and Brown, P. O. (2011). Identification of RNA recognition elements in the Saccharomyces cerevisiae transcriptome. *Nucleic acids research*, 39(4):1501–9.

[127] Robasky, K. and Bulyk, M. L. (2011). UniPROBE, update 2011: expanded content and search tools in the online database of protein-binding microarray data on protein-DNA interactions. *Nucleic Acids Res.*, 39(Database issue).

[128] Rogers, E. and Heitsch, C. E. (2014). Profiling small RNA reveals multimodal substructural signals in a Boltzmann ensemble. *Nucleic acids research*, 42 (22):e171.

[129] Rosenbloom, K. R., Dreszer, T. R., *et al.* (2011). ENCODE whole-genome data in the UCSC Genome Browser: update 2012. *Nucleic Acids Research*.

[130] Saito, H., Yasumoto, K.-I., *et al.* (2002). Melanocyte-specific microphthalmia-associated transcription factor isoform activates its own gene promoter through physical interaction with lymphoid-enhancing factor 1. *J. Biol. Chem.*, 277(32):28787–94.

[131] Schena, M., Shalon, D., *et al.* (1995). Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science*, 270(5235).

[132] Schwarz, D. F., König, I. R., and Ziegler, A. (2010). On safari to random jungle: a fast implementation of random forests for high-dimensional data. *Bioinformatics*, 26(14):1752–1758.

[133] Seo, M. and Oh, S. (2012). CBFS: high performance feature selection algorithm based on feature clearness. *PLoS ONE*, 7(7):e40419.

[134] Shen, N., Zhao, J., *et al.* (2017). Intrinsic specificity differences between transcription factor paralogs partly explain their differential in vivo binding. *bioRxiv*.

[135] Slattery, M., Riley, T., *et al.* (2011). Cofactor binding evokes latent differences in DNA binding specificity between Hox proteins. *Cell*, 147(6):1270–82.

[136] Sokal, R. and Rohlf, F. (2012). *Biometry: the principles and practice of statistics in biological research. 4th edition.* W. H. Freeman and Co.: New York.

[137] Steffen, P., Voss, B., *et al.* (2006). RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 22(4):500–3.

[138] Stormo, G. D. (2000). DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23.

[139] Strobl, C., Boulesteix, A.-L., *et al.* (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, (8):25.

[140] Sylvester, A. M., Chen, D., *et al.* (1998). Role of c-fos and E2F in the induction of cyclin A transcription and vascular smooth muscle cell proliferation. *J. Clin. Invest.*, 101(5):940–8.

[141] Teytelman, L., Thurtle, D. M., *et al.* (2013a). Highly expressed loci are vulnerable to misleading ChIP localization of multiple unrelated proteins. *Proc. Natl. Acad. Sci. USA*, 110(46):18602–7.

[142] Teytelman, L., Thurtle, D. M., *et al.* (2013b). Highly expressed loci are vulnerable to misleading ChIP localization of multiple unrelated proteins. *Proc. Natl. Acad. Sci. USA*, 110(46):18602–7.

[143] Thomas-Chollier, M., Herrmann, C., *et al.* (2012). RSAT peak-motifs: motif analysis in full-size ChIP-seq datasets. *Nucleic Acids Research*, 40(4):e31.

[144] Tinoco, I., Uhlenbeck, O. C., and Levine, M. D. (1971). Estimation of secondary structure in ribonucleic acids. *Nature*, 230(5293):362–7.

[145] Ule, J., Jensen, K. B., *et al.* (2003). CLIP identifies Nova-regulated RNA networks in the brain. *Science*, 302(5648):1212–5.

[146] Waterman, M. and Smith, T. (1978). RNA secondary structure: a complete mathematical analysis. *Math. Biosci.*, 42:257–266.

[147] Watson, J. D. and Crick, F. H. (1953). A Structure for deoxyribose nucleic acid. *Nature*, 171:737–738.

[148] Weintraub, H. (1980). Recognition of specific DNA sequences in eukaryotic chromosomes. *Nucleic acids research*, 8(20):4745–53.

[149] Whitington, T., Frith, M. C., *et al.* (2011). Inferring transcription factor complexes from ChIP-seq data. *Nucleic Acids Research*, 39(15):e98.

[150] Wilbanks, E. G. and Facciotti, M. T. (2010). Evaluation of algorithm performance in ChIP-seq peak detection. *PloS one*, 5(7):e11471.

[151] Wilbert, M. L., Huelga, S. C., *et al.* (2012). LIN28 binds messenger RNAs at GGAGA motifs and regulates splicing factor abundance. *Molecular Cell*, 48(2):195–206.

[152] Workman, C. T., Yin, Y., *et al.* (2005). enoLOGOS: a versatile web tool for energy normalized sequence logos. *Nucl. Acids Res.*, 33(suppl_2):W389–392.

[153] Xiao, W., Adhikari, S., *et al.* (2016). Nuclear m6A Reader YTHDC1 Regulates mRNA Splicing. *Molecular Cell*, 61(4):507–519.

[154] Yao, Z., Weinberg, Z., and Ruzzo, W. L. (2006). CMfinder–a covariance model based RNA motif finding algorithm. *Bioinformatics*, 22(4):445–52.

[155] Yu, L. (2008). Feature Selection for Genomic Data Analysis. In Liu, H. and Motoda, H., editors, *Computational Methods of Feature Selection*, chapter 17, pages 338–354. Chapman & Hall/CRC.

[156] Zhang, Y., Liu, T., *et al.* (2008). Model-based analysis of ChIP-Seq (MACS). *Genome Biol.*, 9(9):R137.

[157] Zhao, Y., Granas, D., and Stormo, G. D. (2009). Inferring binding energies from selected binding sites. *PLoS computational biology*, 5(12):e1000590.

[158] Zuker, M. (1989). On finding all suboptimal foldings of an RNA molecule. *Science (New York, N.Y.)*, 244(4900):48–52.

[159] Zuker, M. (2003). Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research*, 31(13):3406–15.

[160] Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–48.